

Universidad de las Ciencias Informáticas

Facultad 2



Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Título: Sistema de notificaciones vía correo electrónico, mensajería instantánea, SMS y VoIP para el sistema Gestor de Recursos de Hardware y Software (GRHS).

Autor(es):

Elaine de la Caridad Díaz Rodríguez

Víctor Raúl Ramos Linares

Tutor(es):

Ing. Arianna Pérez Carmenates

Ing. Yoanny Torres Rubio

Ing. Rainer Segura Peña

“La Habana 28 de mayo 2016”



“Pero la juventud tiene que crear. Una juventud que no crea es una anomalía realmente.”

DECLARACION DE AUTORIA

Declaramos ser los únicos autores de este trabajo y autorizamos a la Facultad 2 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste, firmamos la presente a los ____ días del mes de _____ del año _____.

Elaine de la C. Díaz Rodríguez

Firma Autor

Víctor Raúl Ramos Linares

Firma Autor

Ing. Yoanny Torres Rubio

Firma Tutor

Ing. Arianna Pérez Carmenates

Firma Tutor

Ing. Rainer Segura Peña

Firma Tutor

DATOS DE CONTACTO

Autor:

Elaine de la Caridad Díaz Rodríguez

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: ecdiaz@estudiantes.uci.cu

Autor:

Víctor Raúl Ramos Linares

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: vrramos@estudiantes.uci.cu

Tutor:

Ing. Arianna Pérez Carmenates

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: acarmenates@uci.cu

Tutor:

Ing. Yoanny Torres Rubio

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: ytrubio@uci.cu

Tutor:

Ing. Rainer Segura Peña

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: rsegura@uci.cu

AGRADECIMIENTOS

Elaine

- ❖ Primero que todo quisiera agradecerles a mis padres Tere y Mario por darme la oportunidad de venir a este mundo, por tanto esfuerzo durante años y por su amor incondicional. Gracias por permitirme ser parte de esa familia tan hermosa que han creado fruto de ese amor tan grande que se tienen, gracias por ser mi ejemplo y mi guía, por apoyarme siempre y estar a mi lado. En especial agradecerle a mi mamá pues gracias a ella hoy me encuentro aquí. Gracias a mi papá por ser mi Pitu, por enseñarme a no rendirme nunca y a luchar siempre por mis sueños. Gracias a ambos por convertirme en esta mujer independiente y fuerte que soy hoy. Los amo mucho.
- ❖ A mi hermano Mario por ser mi compañero de juego y de muchas peleas durante estos diecinueve años, a quien quise desde que estaba en la barriga de mi mamá.
- ❖ A mi primo Jorge Luis por ser un ejemplo y una guía en toda mi vida, por ser más que un primo un hermano mayor. Por decirme siempre la verdad, aunque sea dura.
- ❖ A mi tía Ana por ser más que una tía, una madre. Gracias por siempre ser mi confidente y mi apoyo en todo momento. Gracias por guiarme y enseñarme que no siempre se puede decir lo que uno piensa, porque a veces las verdades duelen.
- ❖ A mi abuela Cuca, por estar estos años a mi lado. Por cuidarme y quererme siempre. Este triunfo es para ti también mima. A mi abuelo Tomás porque a pesar de que no lo recuerdo, sé que me amo muchísimo, y que hoy estaría muy orgulloso de mí.
- ❖ A mis abuelos Libia y Guillermo por ser parte de mi vida. Agradecerle en especial a mi abuelo que, aunque hoy no está a mi lado, sé que se siente orgulloso de mí, por siempre creer en mí y darme fuerzas para seguir adelante.
- ❖ A mi madrina que, aunque hoy este muy lejos sé que está pensando en mí. Por estar a mi lado aun en la distancia, gracias por ser un ejemplo para mí, gracias por quererme tanto.
- ❖ A mi tío Luciano y a mi tío Guillermo por ser parte de mi vida. Gracias a todos mis primos por su amor y su cariño incondicional, en especial a mi primo Julio por ser mi amigo y confidente durante todos estos años, gracias por ser el hombre de mi vida como yo cariñosamente lo llamo, eres muy importante en mi vida. Agradecerle especialmente a mi primo Reinier por estar conmigo estos años de carrera. Gracias a mis tías Lourdes y Nieves, por su confianza y a mi tío Julio por ser el mejor dulcero de la Habana y a mi abuela Juana por quererme tanto.
- ❖ A mis tutores por ayudarnos y guiarnos en este camino de tantos meses, por estar siempre disponibles y dispuestos a ayudarnos.
- ❖ A mi compañero de tesis por su esfuerzo en la realización de este trabajo. Gracias Víctor porque sin ti nada de esto hubiera sido posible.
- ❖ A Mayara (Mi Miji) por ser una persona muy especial en mi vida, por ser más que una amiga, una hermana, por cuidarme y acompañarme siempre en los buenos y malos momentos. Por estar

todos estos años a mi lado, viviendo esta aventura que ha sido nuestro paso por la universidad. Gracias por estar ahí siempre que te necesito, por ayudarme, por apoyarme, por comprenderme, y por sobre todas las cosas por soportarme. Gracias por construir esta bonita amistad que espero nunca termine, eso es lo más grande que me llevo en mi paso por la UCI. Muchas gracias por todo. Espero pronto seas tú quien esté aquí parada defendiendo esto por lo que tanto has luchado.

- ❖ A mis amigos hoy presentes, gracias a todos ustedes. En especial al dúo J++ (Javier y Jessica), a Jimmy, a Adrián, a mis queridas Liz y Yamila (que ya no están aquí), a Lazara y Dania por ser mis amigas más viejas en la UCI, a Yilian, a Yordi, a Rosaibis, a Delia, a Eric, a Manuel y a Stephany, por tantos buenos momentos que compartimos, a Rober y a la flaca d Olga, a Aroldo, a mi tizón querido Pablo que lleva más d 6 años aguantándome y cuidándome. A Lionys por ser alguien muy importante para mí, por ser mí amigo, por escucharme y apoyarme siempre, gracias por todo. En especial agradecer a Yisel y a Boza, por ser mis amigos, por estar siempre conmigo, gracias por cada momento vivido, gracias por ser mi Lalo y mi Te Amo.
- ❖ A Pavón muchas gracias por todo.
- ❖ A Dania la mama de Mayara por quererme como una hija y confiar en mí en momentos difíciles. Gracias por abrirme las puertas de tu casa siempre.
- ❖ A mis compañeros de grupo por ayudarme, por ser parte de mí historia en esta universidad, por los buenos momentos vividos. A todos ustedes gracias.
- ❖ A todos los profesores que intervinieron en mi formación durante estos años, en especial a Omar por tenerme tanta paciencia mientras intentaba enseñarme programación y a su esposa Maidelis. A Ariel por ser un gran profesor de programación. A Héctor por convertirse más que en un profesor, en un buen amigo, gracias por confiar en mí.
- ❖ A Yanet, a Adrián y a Alí por acogerme como parte de su familia y por brindarme tanto cariño. Gracias a los tres por todo lo que hicieron por mí.

Víctor

- ❖ Primero que todo agradecerles a mis padres por su cariño, gracias a la educación que me han dado hoy cumpla esta meta tan importante en mi vida. Por apoyarme y estar siempre a mi lado.
- ❖ A mi esposa por su amor y su compañía durante todo este trayecto. Gracias por todo mi vida te amo mucho.
- ❖ A mis tutores que siempre estuvieron hay pendientes de nosotros y de la tesis en estos meses. Gracias por ayudarnos, guiarnos y sobre todo por regañarnos cuando fue necesario.
- ❖ A mi compañera de tesis, pues sin ella este triunfo no sería posible.
- ❖ A toda mi familia que siempre me apoyo.
- ❖ A mis amigos aquí presentes y a aquellos que no pudieron venir.
- ❖ A los Chacales que me ayudaron y apoyaron siempre.

DEDICATORIA

Elaine

Este trabajo de tantos años y tanto sacrificio va dedicado primero que todo a mis padres, porque a ellos le debo todo lo que soy. Este triunfo es para ustedes, por ser mi motor impulsor y mi guía, por ser mi razón de ser. A mi abuela Cuca y a mi tía Ana por su cariño infinito, por ser mis otras madres. Y a todos los que me ayudaron de una forma u otra a conseguir este triunfo.

Víctor

Le dedico esta tesis a mi mamá y a mi papá porque siempre estuvieron hay para darme ánimos y a poyarme en todo el transcurso de la carrera y a mi esposa que, aunque no ha sido fácil para ella tenerme lejos me ha apoyado y ayudado incondicionalmente.

RESUMEN

El Centro de Telemática (TLM) de la Universidad de las Ciencias Informáticas (UCI) se encarga del desarrollo de sistemas y servicios informáticos en la rama de Telecomunicaciones y Seguridad Informática, actualmente cuenta con el Sistema Gestor de Recursos de Hardware y Software (GRHS) encargado fundamentalmente de obtener la información de las estaciones de trabajo presentes en la entidad. Dentro de sus funcionalidades se destaca la detección de incidencias en los clientes, tanto de hardware como de software, en este momento estas solo se notifican al personal interesado vía correo electrónico. Con el avance de las nuevas tecnologías y el surgimiento de nuevos mecanismos de envío de notificaciones surge la necesidad de incorporar nuevos métodos para realizar tan importante labor, en aras de garantizar la rápida respuesta a las incidencias generadas. El presente trabajo consiste en la creación de un sistema que permita realizar el envío de notificaciones desde GRHS a través de correo electrónico, mensajería instantánea, SMS y VoIP, logrando que existan diferentes vías para el envío de las notificaciones permitiendo el control de los recursos de hardware y software del centro, siendo de vital importancia en la toma de decisiones.

Palabras claves: incidencias, notificaciones, vías de comunicación.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA	4
1.1 Conceptos.....	4
1.1.1 Notificaciones	4
1.1.2 Correo electrónico	4
1.1.3 Mensajería instantánea (IM)	4
1.1.4 Servicio de mensajes cortos (SMS)	4
1.1.5 Voz sobre el protocolo de Internet (VoIP)	5
1.1.6 Plugin	5
1.2 Sistemas integrados para el envío de notificaciones a nivel internacional.....	5
1.2.1 Facebook Notifications (Notificaciones de Facebook).....	5
1.2.2 MySpace	6
1.2.3 AlertFind.....	6
1.3 Sistemas integrados para el envío de notificaciones a nivel nacional.....	7
1.3.1 Sistema integrado para el envío de notificaciones vía correo electrónico, mensajería instantánea, SMS y llamadas VoIP.....	7
1.3.2 Sistema de Integración de Servicios de Comunicación a Usuarios (SISCU).....	7
1.4 Conclusiones del estudio de sistemas para el envío de notificaciones a nivel internacional y nacional.	7
1.5 Metodología de desarrollo de software.	8
1.5.1 Metodología de desarrollo seleccionada: Extreme Programming (XP).	8
1.6 Herramientas y Tecnologías	9
1.6.1 Lenguaje de Programación.....	9
1.6.2 Framework de desarrollo	10
1.6.3 IDE de desarrollo.....	11
1.6.4 Herramienta CASE	11
1.6.5 Sistema Gestor de Bases de Datos (SGBD).....	12
1.7 Conclusiones parciales	12
CAPÍTULO 2 PROPUESTA DE SOLUCIÓN	13

2.1 Propuesta de solución	13
2.1.1 Gestión de notificaciones.....	14
2.1.2 Servicio de envío de notificaciones.....	14
2.2 Lista de funcionalidades.....	15
2.3 Características del sistema.....	17
2.3.1 Usabilidad	17
2.3.2 Eficiencia.....	17
2.3.3 Software	17
2.3.4 Interfaz	18
2.4 Fase de Exploración	18
2.4.1 Historias de usuarios	18
2.5 Fase de planificación	23
2.5.1 Estimación del esfuerzo de cada historia de usuario	24
2.5.2 Plan de iteraciones	25
2.5.3 Plan de duración de iteraciones.....	26
2.5.4 Plan de entregas	28
2.6 Conclusiones parciales	28
CAPÍTULO 3 DISEÑO, IMPLEMENTACIÓN Y PRUEBA	29
3.1 Arquitectura de Software.....	29
3.1.1 Arquitectura Cliente-Servidor.....	29
3.2 Patrón arquitectónico	30
3.2.1 Modelo-Vista-Plantilla.....	30
3.3 Patrones de diseño	31
3.3.1 Patrón GRASP	31
3.4 Tarjeta Clase –Responsabilidad –Colaborador (CRC)	34
3.5 Diagrama de clases persistentes	35
3.6 Modelo físico de la Base de Datos.....	35
3.7 Tareas de Ingeniería.....	36
3.8 Pruebas	39
3.9 Conclusiones parciales	46

CONCLUSIONES GENERALES	48
RECOMENDACIONES	49
REFERENCIAS BIBLIOGRÁFICAS	50
BIBLIOGRAFIA	52
ANEXO 1.....	55
ANEXO 2.....	61
ANEXO 3.....	63

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Subsistemas del sistema de notificaciones.....	13
Ilustración 2: Arquitectura Cliente-Servidor.....	29
Ilustración 3: Patrón arquitectónico: Modelo –Plantilla –Vista	30
Ilustración 4: Clase urls.py	32
Ilustración 5: Clase Notificaciones	33
Ilustración 6: Clase CorreoSerializer	33
Ilustración 7: Diagrama de clases persistentes.....	35
Ilustración 8: Modelo físico de la base de datos	36
Ilustración 9: Prueba unitaria al sistema de envío de notificaciones.	39
Ilustración 10: Resultados de las pruebas de aceptación.	46

ÍNDICE DE TABLAS

Tabla 1: HU # 1 Crear notificación.....	19
Tabla 2: HU # 2 Listar notificaciones.	20
Tabla 3: HU # 3 Eliminar notificación.....	21
Tabla 4: HU # 4 Detalles de la notificación.	22
Tabla 5: Estimación de esfuerzo por HU.	24
Tabla 6: Plan de duración de las iteraciones.	26

Tabla 7: Plan de entregas	28
Tabla 8: CRC. Clase: Correo.....	34
Tabla 9: CRC. Clase: CorreoSerializer.....	34
Tabla 10: CRC. Clase: CorreoMainMenu.	34
Tabla 11 CRC. Clase: CorreoMenu.....	34
Tabla 12: Tarea de Ingeniería # 1 Crear notificación.....	37
Tabla 13: Tarea de Ingeniería # 2 Validar notificación.....	37
Tabla 14: Tarea de Ingeniería # 3 Listar notificaciones.....	38
Tabla 15: Tarea de Ingeniería # 4 Eliminar notificación.....	38
Tabla 16: Caso de Prueba Eliminar notificación.....	40
Tabla 17: Caso de Prueba Adicionar configuración de correo electrónico.....	41
Tabla 18: Caso de Prueba Adicionar configuración de mensajería instantánea.....	42
Tabla 19: Caso de Prueba Adicionar configuración de SMS.....	43
Tabla 20: Caso de Prueba Adicionar configuración de VoIP.....	45
Tabla 21: HU # 5 Listar tipo de notificación.....	55
Tabla 22: HU # 6 Adicionar tipo de notificación.....	55
Tabla 23: HU # 7 Editar tipo de notificación.....	56
Tabla 24: HU # 8 Eliminar tipo de notificación.....	57
Tabla 25: HU # 9 Listar configuración de correo electrónico.....	58
Tabla 26: HU # 10 Adicionar configuración de correo electrónico.....	59
Tabla 27 CRC. Clase: Jabber.....	61
Tabla 28 CRC. Clase: JabberSerializer.....	61
Tabla 29 CRC. Clase: JabberMainMenu.....	61

Tabla 30 CRC. Clase: JabberMenu.....	61
Tabla 31 CRC. Clase: SMS.....	62
Tabla 32 CRC. Clase: SMSSerializer.....	62
Tabla 33 CRC. Clase: SMSMainMenu.....	62
Tabla 34: Tarea de Ingeniería # 5 Detalles de la notificación.....	63
Tabla 35: Tarea de Ingeniería # 6 Listar tipo de notificación.....	63
Tabla 36: Tarea de Ingeniería # 7 Adicionar tipo de notificación.....	63
Tabla 37: Tarea de Ingeniería # 8 Editar tipo de notificación.....	64
Tabla 38: Tarea de Ingeniería # 9 Eliminar tipo de notificación.....	64
Tabla 39: Tarea de Ingeniería # 10 Listar configuración de correo electrónico.....	65
Tabla 40: Tarea de Ingeniería # 11 Adicionar configuración de correo electrónico.....	65
Tabla 41: Tarea de Ingeniería # 12 Validar configuración de correo electrónico.....	66

INTRODUCCIÓN

El hombre como ser social siempre sintió la necesidad de comunicarse con sus semejantes, por ello se dio a la tarea de crear medios que permitieran realizar dicha acción. Con el transcurso de los años fueron innumerables los avances que el hombre llevó a cabo en el área de las comunicaciones, con el objetivo de transmitir información a una mayor cantidad de personas que se encontraban a grandes distancias en un corto periodo de tiempo, de ahí que surgieran innovadores inventos como el telégrafo, el teléfono, y años después la telefonía móvil y el Internet.

Internet ha sido sin lugar a duda uno de los mayores avances en el vertiginoso mundo de las TIC¹. Su creciente desarrollo ha dado paso al surgimiento de nuevas vías de comunicación como correo electrónico, mensajería instantánea, SMS² y VoIP³ que permiten la comunicación de dos o más individuos que se encuentran a grandes distancias. Hoy día uno de los servicios más difundidos es el correo electrónico, el cual permite a los usuarios enviar y recibir mensajes o cartas electrónicas, además de imágenes, audio y videos. La mensajería instantánea en la actualidad ha aumentado en gran medida su popularidad entre los usuarios debido a que admite el establecimiento de conversaciones entre dos o más usuarios en tiempo real, posibilitando la existencia de una comunicación constante. Otro servicio de gran demanda es el SMS, este servicio permite a los usuarios el enviar y recibir mensajes cortos desde sus teléfonos móviles. También existe la transmisión de datos usando el sistema de telefonía, en este caso el término utilizado es VoIP, que es utilizado en telefonía IP⁴ para definir los servicios que se usan para transmitir voz usando el protocolo IP.

En un mundo cada vez más dependiente de la tecnología donde las personas se relacionan constantemente usando medios electrónicos resulta vital tener retroalimentación de todos los procesos asociados al quehacer en el ciberespacio, es por ello que redes sociales, sitios de noticias o software de control empresarial se preocupen por notificar a sus usuarios de los eventos que resultan de su interés. Ejemplo de estas notificaciones son las solicitudes de amistad, mensajes de amigos, las invitaciones a eventos, y la promoción sobre la venta de nuevos productos en páginas comerciales.

El Centro de Telemática (TLM) es uno de los centros de desarrollo de software de la Universidad de Ciencias Informáticas (UCI), y su trabajo se centra en el desarrollo de sistemas y servicios informáticos en la rama de telecomunicaciones y seguridad informática. Uno de los sistemas desarrollados en el centro es el Sistema Gestor de Recursos de Hardware y Software (GRHS), sistema encargado de obtener las

¹ TIC: Tecnologías de la Información y las Comunicaciones.

² SMS: Short Message Service o Servicio de Mensajes Cortos.

³ VoIP: Voice over Internet Protocol o Voz sobre el Protocolo de Internet.

⁴ IP: Internet Protocol o Protocolo de Internet.

propiedades de hardware⁵ y software⁶ en una red de computadoras, con el fin de contabilizar los cambios realizados en cada una de las estaciones de trabajo y ejecutar acciones automáticas en caso de cambios no autorizados, a estos cambios no autorizados se les llama incidencias. Dentro de sus funcionalidades se destaca la detección de incidencias de los clientes, tanto de hardware como de software, siendo el responsable del sistema quien designa que acción es considerada una y cual no, dándole diferentes prioridades según su relevancia. En el caso de hardware es considerada una incidencia la conexión o desconexión de periféricos, siendo consideradas de alta prioridad, y en cuanto al software la instalación o desinstalación de alguno de estos en las computadoras.

Actualmente el sistema envía notificaciones a través del correo electrónico, este es el único medio de comunicación que se utiliza para el envío de dichas notificaciones, lo que ocasiona que los responsables del sistema cuenten con un solo medio para esto. Esta situación aumenta el riesgo de que el problema tarde en resolverse o que ocurra la pérdida de algún dispositivo de las estaciones de trabajo. Otra de las deficiencias que trae consigo la utilización de esta única vía de comunicación es que, de existir algún inconveniente con el servidor de correo electrónico, las notificaciones pudieran no ser recibidas, trayendo consigo un problema en la toma de decisiones.

Teniendo en cuenta lo expuesto anteriormente se plantea el siguiente **problema a resolver**: ¿Cómo gestionar las notificaciones asociadas al Sistema Gestor de Recursos Hardware y Software (GRHS)?

Se define como **objeto de estudio** los procesos asociados al envío de notificaciones.

Para dar solución al problema a resolver se define como **objetivo general**: desarrollar un sistema que permita gestionar las notificaciones del Sistema Gestor de Recursos Hardware y Software (GRHS).

El **campo de acción** se centra en el proceso de gestión de notificaciones del Sistema Gestor de Recursos Hardware y Software (GRHS).

Para dar cumplimiento al objetivo propuesto se organizan las siguientes **tareas de investigación**:

- 1- Estudio de sistemas informáticos basados en el envío de notificaciones para analizar sus características.
- 2- Análisis del sistema GRHS para identificar los mecanismos de envío de notificaciones.
- 3- Análisis de la tecnología, metodología y herramientas a emplear para una correcta implementación del sistema de envío de notificaciones.
- 4- Análisis de las pruebas propuestas por la metodología para la futura validación del sistema desarrollado.

⁵ Conjunto de elementos físicos o materiales que constituyen una computadora o un sistema informático.

⁶ Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas.

Para el desarrollo de esta investigación y el logro de su objetivo, se utilizaron los siguientes **métodos de investigación científica**:

- **Métodos Teóricos:**

- **Analítico sintético:** Este método permitió analizar las teorías y los documentos referentes al objetivo de la investigación, facilitando de esta forma la extracción de los elementos más importantes relacionados con el objeto de estudio.
- **Modelación:** Este método se utilizó para la realización de los modelos y diagramas presentados en la aplicación a desarrollar. Los diagramas y modelos a realizar se hacen mediante la herramienta Visual Paradigm utilizando UML como lenguaje de modelado.

- **Métodos Empíricos:**

- **Entrevista:** Este método se utilizó para el desarrollo del sistema, detectando las necesidades del cliente a través del intercambio con él. Se realizan entrevistas con el cliente para poder satisfacer sus necesidades y obtener un software con la calidad requerida.

El documento de investigación se divide en 3 capítulos, los cuales están estructurados de la siguiente forma:

- Capítulo 1. “**Fundamentación teórica**”: se describen los principales conceptos a tratar. Se realizó la investigación y el estudio del estado del arte, analizando los sistemas similares que permitan llevar a cabo el proceso de envío de notificaciones vía correo electrónico, mensajería instantánea, SMS y VoIP. Se definió la metodología de desarrollo de software, herramientas y tecnologías a utilizar para la implementación del sistema.
- Capítulo 2. “**Propuesta de solución**”: se describe la propuesta de solución creada por los desarrolladores, además se profundiza en las fases de exploración y planificación planteadas por la metodología XP, planteándose a grandes rasgos las Historias de Usuarios (HU) y se realiza una propuesta de los prototipos no funcionales de la aplicación.
- Capítulo 3. “**Diseño, implementación y prueba**”: se especifica la arquitectura del sistema y con ello los patrones arquitectónicos empleados. Se adapta el diseño al entorno de implementación y se describen las tareas de ingeniería. Se explicarán las pruebas realizadas al software para comprobar que el producto de software funciona según lo diseñado y que las funcionalidades se han implementado de forma adecuada.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

En este capítulo se explican los conceptos más importantes relacionados con el desarrollo del trabajo de diploma. Se lleva a cabo un estudio que enmarca las diferentes herramientas a nivel internacional y nacional que hacen uso del envío de notificaciones. Se especifica la metodología de desarrollo de software escogida según las características del equipo de desarrollo, así como las tecnologías, herramientas y lenguajes utilizados para la implementación del sistema.

1.1 Conceptos

Con el estudio realizado se han encontrado algunos conceptos relacionados con el campo de acción, que son de vital importancia para un mejor entendimiento del contenido de la investigación. A continuación, se presentan los más importantes.

1.1.1 Notificaciones

Según la Real Academia de La Lengua Española, “las notificaciones son acciones o efectos de comunicar. Siendo los sistemas de notificaciones aplicaciones que generan y envían mensajes a sus usuarios por un canal específico, manteniéndolos informados en todo momento”.(1)

1.1.2 Correo electrónico

El correo electrónico es un servicio de red que permite a los usuarios enviar y recibir mensajes en fracciones de segundos, con textos, sonidos e imágenes. El emisor envía los mensajes a un servidor y este, a su vez, se encarga de enviarlos al servidor del receptor. Para poder ver el correo electrónico es necesario que el receptor se conecte con su servidor. Se puede enviar el mensaje a uno o varios remitentes al mismo tiempo, con dirección visible o encriptado, con listas de distribución públicas o privadas.(2)

1.1.3 Mensajería instantánea (IM)

Sistema de mensajería consistente en el intercambio de mensajes de texto en tiempo real entre dos o más personas; es más interactivo que el correo electrónico, ya que los mensajes se envían inmediatamente, mientras que los mensajes de correo electrónico pueden estar en cola de espera en un servidor de correo durante unos segundos o minutos.(3)

1.1.4 Servicio de mensajes cortos (SMS)

Conocido por la sigla SMS, con este servicio se puede enviar mensajes de texto desde cualquier móvil o aplicación a otro, sin importar de qué compañía sea. Si el teléfono al que se envía el mensaje está apagado o fuera de cobertura el mensaje se almacena en la red y se entrega en cuanto el teléfono se conecta de nuevo a la red.(4)

1.1.5 Voz sobre el protocolo de Internet (VoIP)

VoIP es el acrónimo de "Voice Over Internet Protocol", que tal y como el término dice, hace referencia a la emisión de voz en paquetes IP sobre redes de datos como puede ser Internet. Llegados a este punto se unen dos mundos que hasta entonces habían convivido separados: la transmisión de voz y la de datos. La tecnología VoIP trata de transportar la voz, previamente procesada, encapsulándola en paquetes para poder ser transportada sobre redes de datos sin necesidad de disponer de una infraestructura telefónica convencional. Con lo que se consigue desarrollar una única red homogénea en la que se envía todo tipo de información ya sea voz, video o datos.(5)

1.1.6 Plugin

Un plugin (o plug-in en inglés "enchufar", también conocido como addin, add-in, addon o add-on) es una aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica, como por ejemplo servir como driver⁷ en una aplicación, para hacer así funcionar un dispositivo en otro programa. Esta aplicación adicional es ejecutada por la aplicación principal. Los plugins típicos tienen la función de reproducir determinados formatos de gráficos, reproducir datos multimedia, codificar/decodificar emails, filtrar imágenes de programas gráficos.(6)

1.2 Sistemas integrados para el envío de notificaciones a nivel internacional.

1.2.1 Facebook Notifications (Notificaciones de Facebook)

Es un pequeño programa gratuito que permite tener las notificaciones de Facebook directamente en el escritorio del usuario. Permitiendo que este reciba dichas notificaciones, aun cuando no tenga abierta su cuenta, ni el correo para que le comunique que se tiene una nueva notificación. Esta aplicación indica si se tienen nuevos mensajes, eventos e invitaciones, además de que permite actualizar el estado desde la aplicación.(7)

Entre los eventos que permite notificar esta aplicación se encuentran los que se relacionan a continuación:

- Algún amigo cambia su estado o su perfil.
- Te envían una petición de amigo, evento o grupo.
- Alguien escribe en el muro de alguno de tus amigos.
- Una foto es etiquetada con el nombre de uno de tus amigos.

Facebook envía diferentes tipos de notificaciones en dependencia del sistema que esté utilizando el usuario:

⁷ Componente de software que permite que un dispositivo se entienda con el sistema operativo y pueda ser utilizado por las aplicaciones.

- Si el usuario está utilizando una computadora o un celular, Facebook hace uso de las siguientes notificaciones:
 - Notificaciones de alerta roja: números de color rojo que aparecen sobre los iconos en la parte superior derecha del sitio web.
 - Notificaciones por correo electrónico: notificaciones que recibe el usuario por correo electrónico.
- Si el usuario está utilizando solo una computadora, Facebook hace uso de las notificaciones emergentes, estas son notificaciones que aparecen en la pantalla mientras se mantenga una sesión iniciada en Facebook.
- Si el usuario está utilizando un celular, Facebook hace uso de las siguientes notificaciones:
 - Notificaciones por SMS: notificaciones recibidas a través de un mensaje de texto.
 - Notificaciones push: notificaciones que aparecen en los dispositivos móviles cuando no se está utilizando Facebook.(8)

1.2.2 MySpace

MySpace es un sitio web de interacción social, con sede central en Beverly Hills, California. Se convirtió en la red social más popular en Estados Unidos de América en junio del 2006. Una de las aplicaciones con que cuenta MySpace hoy en día es con MySpace Notifier 1.3, que es una aplicación de escritorio donde se le notifica al usuario de los eventos que ocurren en la página web. Esta aplicación se instala como un icono más de la bandeja de sistema donde avisa al usuario cada vez que su espacio MSN recibe nuevos comentarios, mensajes y pedidos de amigos. El programa muestra una ventana emergente tipo balón y su icono cambia para cada notificación. Una vez recibida la notificación, se da clic en la ventana de notificación y se trasladará automáticamente a la cuenta del usuario que lo está solicitando. Requiere para su uso, utilizar como navegador el Internet Explorer 4.0 o superior.(9)

1.2.3 AlertFind

Es un software para el envío de notificaciones que la compañía Dell Incorporated brinda por suscripción. Es un sistema automatizado, un servicio de notificación de esta empresa basado en la nube, respaldado por su compromiso con la calidad y la fiabilidad. Independiente de su infraestructura, los clientes de AlertFind pueden enviar notificaciones a cientos o miles de empleados a través de correo electrónico, teléfono fijo, teléfono móvil y mensajes de texto SMS, permitiéndoles responder dichas notificaciones(10).

La utilización de este software no se basa solo en el envío de información, sino que también permite recoger las respuestas de los usuarios, permitiendo establecer una comunicación bidireccional. A medida que se envían las notificaciones, el software proporciona alertas y respuestas tablero,

mostrando el estado en tiempo real de las notificaciones y reconocimientos con capacidad de desglose para más detalles.

1.3 Sistemas integrados para el envío de notificaciones a nivel nacional.

1.3.1 Sistema integrado para el envío de notificaciones vía correo electrónico, mensajería instantánea, SMS y llamadas VoIP.

Este sistema fue desarrollado en el año 2013 bajo la autoría de la Universidad de las Ciencias Informáticas como resultado de un trabajo de diploma. Tiene el propósito de permitir la conexión de aplicaciones clientes y que estas a su vez puedan enviar notificaciones, alertas, mensajes, noticias y reportes, utilizando diferentes medios de comunicación como correo electrónico, mensajería instantánea, SMS y llamadas VoIP u otros que se deseen adicionar. La aplicación es capaz de permitir la comunicación con los servidores o dispositivos de estos servicios independientemente de las tecnologías que se utilicen en el desarrollo de las aplicaciones, integrando el envío de las notificaciones por las vías de comunicación antes mencionadas.

1.3.2 Sistema de Integración de Servicios de Comunicación a Usuarios (SISCU).

El Sistema de Integración de Servicios de Comunicación a Usuarios (SISCU) es un software creado en la Universidad de las Ciencias Informáticas (UCI) con el propósito de disminuir el tiempo y esfuerzo de los usuarios al enviar mensajes. Las notificaciones pueden ser enviadas por los diferentes medios de comunicación interpersonales existentes en la UCI como correo electrónico, mensajería instantánea, telefonía fija y SMS.

El sistema es una aplicación web⁸ a la cual puede acceder cualquier usuario previamente autenticado. Permite el envío de mensajes y brinda la posibilidad de gestionar los clientes y contactos que posee cada usuario. Esto facilita el trabajo de los mismos, ya que no tendrán que utilizar cada una de las aplicaciones que posibilitan el envío de mensajes de forma individual, sino que se presentan todas en un único sistema. Actualmente el sistema está en desuso.

1.4 Conclusiones del estudio de sistemas para el envío de notificaciones a nivel internacional y nacional.

El estudio realizado mostró la existencia de varios sistemas que integran el envío de notificaciones mediante diferentes vías de comunicación (Facebook Notifications, MySpace, AlertFind, Sistema integrado para el envío de notificaciones vía correo electrónico, mensajería instantánea, SMS y llamadas VoIP). Estas aplicaciones poseen un conjunto de características que se pueden tener en cuenta para el desarrollo del sistema como son la integración de diferentes vías de comunicación para el envío de

⁸ Aplicación web: aplicación que sirve procedimientos y páginas web a través de un sistema de comunicaciones y que suele utilizar como protocolo de comunicaciones el HTTP.

notificaciones y la gestión de usuarios, sin embargo, no pueden ser utilizados para la solución de la problemática existente en el presente trabajo de diploma, puesto que:

- Facebook Notifications, MySpace y AlertFind son sistemas que están diseñados con el propósito de servir a las necesidades específicas de las aplicaciones que los usan; además de que no admiten la integración de nuevos servicios de comunicación al sistema, ya que estos están predefinidos. En el caso de AlertFind no realiza el envío de notificaciones vía mensajería instantánea, en su lugar Facebook Notifications hace uso de la mensajería instantánea en cambio no utiliza el envío de notificaciones vía VoIP.
- SISCO y Sistema integrado para el envío de notificaciones vía correo electrónico, mensajería instantánea, SMS y llamadas VoIP no pueden ser utilizados en el presente trabajo debido a que no se cuenta con los productos resultantes de estas investigaciones. Además, SISCO es un sistema que no permite la comunicación con aplicaciones clientes para el envío de notificaciones, además de que no permite la integración de nuevos servicios de comunicación ya que los mismos están predefinidos.

Las razones antes expuestas exaltan la necesidad de desarrollar un sistema que permita el envío de notificaciones para el sistema GRHS. Las evidencias muestran que no es factible la utilización de estas aplicaciones debido a temas relacionados con el carácter privativo del software y/o la inexistencia del producto final de las investigaciones o código fuente del mismo. Estos elementos atentan contra la posibilidad de reutilización de dichas aplicaciones, siendo el desarrollo de un nuevo sistema para el envío de notificaciones la mejor variante. La confección de este nuevo producto sería consecuente con las políticas del país sobre la utilización y desarrollo de aplicaciones y tecnologías libres, y respondería a las verdaderas necesidades y particularidades del entorno del sistema GRHS.

1.5 Metodología de desarrollo de software.

1.5.1 Metodología de desarrollo seleccionada: Extreme Programming (XP).

Es la más destacada de los procesos ágiles de desarrollo de software. La programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad, donde se considera que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos.(11) Promueve el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo.

Para la realización del presente trabajo de diploma se decide emplear la metodología ágil XP. A continuación, se exponen las razones de la elección:

El equipo de trabajo está compuesto por dos programadores, lo cual disminuye el número de errores durante la programación, donde el grado de interacción entre los miembros del equipo es alto, pues

existe una gran compenetración y un buen entendimiento entre ellos. Este elemento permite que el código se pueda reescribir sin modificar su comportamiento para así hacerlo lo más simple y legible posible. Los requisitos del proyecto pueden cambiar, pues el cliente puede agregar nuevas funcionalidades según avanza el proyecto, siendo la generación de artefactos y la documentación muy poca, ya que el tiempo de desarrollo del proyecto es corto. El cliente forma parte del equipo de desarrollo, lo cual ayuda a corregir a tiempo los errores que puedan surgir.

1.6 Herramientas y Tecnologías

Las herramientas y tecnologías seleccionadas para desarrollar un proyecto deben ser escogidas cuidadosamente, ya que estas podrían suponer el fracaso del proyecto o podrían ser las causantes del aumento de la complejidad del mismo, por ello resulta vital saber cuáles son las diferentes alternativas existentes y cuáles son las necesidades específicas del proyecto a desarrollar. Luego de ser analizadas las ideas anteriores en consecuencia con el tipo de proyecto que se desea desarrollar se definieron un conjunto de herramientas y tecnologías válidas, para llevar a cabo tal tarea. Las herramientas y tecnologías seleccionadas para la creación del sistema de envío de notificaciones vía correo electrónico, mensajería instantánea, SMS y VoIP para GRHS son las siguientes:

1.6.1 Lenguaje de Programación

Un lenguaje de programación permite crear un grupo de instrucciones que luego se convertirán en un programa, para el desarrollo del sistema será utilizado Python 2.7 como lenguaje del lado del servidor y JavaScript 1.9.1 como lenguaje del lado del cliente.

1.6.1.1 Python 2.7

Python es un lenguaje de programación interpretado, multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional, además de que permite al desarrollador trabajar con mayor rapidez e integrar sus sistemas con mayor eficacia.(12) Se utiliza en conjunto con el framework Django 1.4, ya que el mismo permite la integración de numerosas librerías que facilitarán la implementación de algunas funcionalidades del sistema.

1.6.1.2 JavaScript 1.9.1

JavaScript es un lenguaje de programación, se utiliza principalmente del lado del cliente permitiendo crear efectos atractivos y dinámicos en las páginas web. Los navegadores modernos interpretan el código JavaScript integrado en las páginas web.(13) Se utiliza este lenguaje para validar los datos de entrada en las interfaces del sistema.

1.6.2 Framework de desarrollo

Un framework de desarrollo es una estructura conceptual y tecnológica de soporte definido, además de ser un ambiente de trabajo que contiene librerías de códigos y módulos que pueden ser reutilizados para el rápido desarrollo de aplicaciones. También suele incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas. Para el desarrollo del sistema se utiliza Django 1.4.

1.6.2.1 Django 1.4

Django es uno de los frameworks de código abierto más importante para el desarrollo web, es un entorno de desarrollo web escrito en Python que presume de construir aplicaciones web más rápido y con menos código. Django 1.4 incluye entre sus características el soporte de zonas horarias, mejoras en seguridad gracias al almacenamiento de contraseñas utilizando códigos hash, integración con frameworks de testing en el navegador o soporte de Python 2.7.(14)

1.6.2.2 jQuery 1.9

jQuery es una biblioteca gratuita de JavaScript, cuyo objetivo principal es simplificar las tareas de creación de páginas web responsables, acordes a lo estipulado en la web 2.0, la cual funciona en todos los navegadores modernos(15). Se utiliza el framework de desarrollo jQuery en su versión 1.9 ya que ayuda de gran manera en el diseño del sitio, al abstraer por completo todas las características específicas de cada uno de los navegadores y se enfoca en simplificar los scripts y en acceder o modificar el contenido de una página web.

1.6.2.3 TwitterBootstrap 3.0

Bootstrap es un framework que simplifica el proceso de creación de diseños web combinando CSS⁹ y JavaScript. Es un framework potente con numerosos componentes web que ahorra mucho esfuerzo y tiempo. Bootstrap se divide en:

- ✓ **Scaffolding:** estructura agradable con estilos globales, grids, variables, además utiliza elementos HTML¹⁰ y propiedades CSS que necesitan de un Doctype HTML 5
- ✓ **CSS:** dispone de un conjunto de clases para aplicar tanto al formateo de código como a las tablas, formularios, botones, tipografía e incluso íconos.
- ✓ **Componentes:** para ofrecer mayor interactividad al sitio cuenta con componentes como grupos de botones, alertas, navegación y tipografía.
- ✓ **Plugins de JavaScript:** brinda la posibilidad de crear plugins de JQuery personalizados

⁹ CSS: Cascading Style Sheets (Hojas de Estilo en Cascada).

¹⁰ HTML: HyperText Markup Language (lenguaje de marcas de hipertexto)

- ✓ **LESS:** para optimizar el trabajo con el CSS. Da la oportunidad de extender las posibilidades de las hojas de CSS normales.(16)

En el desarrollo de la aplicación se emplea Bootstrap 3.0 para la creación de las interfaces, ofreciendo una mejor integración con librerías como JQuery.

1.6.2.4 Backbone 1.0

Backbone es un pequeño framework que permite construir aplicaciones usando JavaScript. Su objetivo consiste en probar y definir un conjunto de estructuras de datos junto al manejo de la interfaz por medio de vistas que son útiles cuando se construyen aplicaciones JavaScript.(17)

1.6.3 IDE de desarrollo

El Entorno de Desarrollo Integrado (IDE por sus siglas en inglés Integrated Development Environment), es un programa compuesto por un conjunto de herramientas de programación útiles para el desarrollador de software, con el objetivo de mejorar la productividad de este y obtener mayor rapidez en el desarrollo. Para la implementación del sistema se utiliza PyCharm ya que permite desarrollar en el lenguaje de programación Python, y tiene soporte para el framework de desarrollo Django.

1.6.3.1 PyCharm 4.5.1

PyCharm es un IDE o entorno de desarrollo integrado multiplataforma utilizado para desarrollar en el lenguaje de programación Python. Proporciona análisis de código, depuración gráfica, integración con VCS / DVCS y soporte para el desarrollo web con Django, entre otras bondades. PyCharm es desarrollado por la empresa JetBrains y debido a la naturaleza de sus licencias tiene dos versiones, la Community que es gratuita y orientada a la educación y al desarrollo puro en Python y la Professional(18).

1.6.4 Herramienta CASE

Las herramientas de Ingeniería de Software Asistida por Computadora (CASE, por sus siglas en inglés) constituyen un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un sistema.(19)

1.6.4.1 Visual Paradigm 8.0

Dentro de la familia de las herramientas CASE, Visual Paradigm es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML¹¹ ayuda a una rápida construcción de aplicaciones de mejor calidad con un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.(20)

¹¹ Lenguaje Unificado de Modelado

1.6.5 Sistema Gestor de Bases de Datos (SGBD)

Un sistema gestor de bases de datos (SGBD) es un conjunto de programas que permiten la creación de base de datos y proporciona herramientas para añadir, borrar, modificar y eliminar los elementos de la misma, además de mantener la integridad, confidencialidad y seguridad de los mismos.

1.6.5.1 PostgreSQL 9.3

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente.

PostgreSQL tiene las siguientes características principales:

- Arquitectura cliente-servidor con un amplio rango de drivers y clientes.
- Diseño de alta concurrencia donde los lectores y escritores no se bloquean.
- Altamente configurable y extensible para muchos tipos de aplicación.
- Excelente escalabilidad y rendimiento con características de ajustes extensas.
- Optimizador de consultas sofisticado, adecuado para inteligencia de negocios.
- Soporta totalmente el acceso y procedimientos de base de datos en Java, Python, Perl, PHP y muchos más.
- Altamente confiable con características extensivas para durabilidad y alta disponibilidad.
- Soporta el uso de índices, reglas y vistas.
- Incluye herencia entre tablas aunque no entre objetos, ya que no existen.(21)

1.7 Conclusiones parciales

Con el objetivo de darle cumplimiento al objetivo general y a la problemática planteada en el presente trabajo, se cumplieron algunas de las tareas de investigación planteadas de manera satisfactoria en el presente capítulo. Por lo tanto, se concluye que:

- ✓ El estudio realizado de las herramientas informáticas existentes arrojó que no es posible el uso de estas en la universidad ya que no cumplen con todos los requerimientos del sistema que se pretende implementar.
- ✓ La utilización de la metodología ágil XP es adecuada para el desarrollo del software, ya que se adapta a las necesidades del mismo por las características del sistema a implementar, permitiendo guiar el proceso de desarrollo del software de forma correcta.
- ✓ Las herramientas y tecnologías fueron escogidas adecuadamente para desarrollar la aplicación, teniendo en cuenta las particularidades de cada una de estas y las características del sistema a implementar.

CAPÍTULO 2 PROPUESTA DE SOLUCIÓN

En este capítulo se describirán las características del sistema a desarrollar. Se hace alusión a las fases de exploración y planificación propias de la metodología de desarrollo que se utilizará en el desarrollo del sistema. Se describe la lista de funcionalidades con que debe contar el sistema y las características del mismo. Se realiza el plan de duración de las iteraciones a llevar a cabo y el plan de entregas donde se detalla en qué momento se hará entrega de las diferentes versiones del sistema.

2.1 Propuesta de solución

La solución propuesta está diseñada para permitir el envío de notificaciones vía correo electrónico, mensajería instantánea, SMS y VoIP de manera automatizada, permitiendo así que los directivos del centro de TLM reciban las notificaciones con la mayor inmediatez posible. Esto será posible debido a que la información llegará por cuatro vías diferentes ayudando al control de los recursos de hardware y software del centro, siendo de vital importancia en la toma de decisiones. El producto resultante de esta investigación se plantea como un sistema independiente que permite la integración con el sistema GRHS, ambos sistemas están desarrollados bajo las mismas tecnologías, permitiendo así que la integración de estos sea menos compleja.

La solución propuesta permite dotar al sistema GRHS de múltiples vías para realizar el envío de las notificaciones de incidencias, posibilitando a su vez que la información llegue de forma adecuada y en el momento preciso a los encargados de la toma de decisiones en cualquier entidad que haga uso de este sistema.

El sistema de envío notificaciones estará compuesto por dos subsistemas, como se muestra en la ilustración 1:

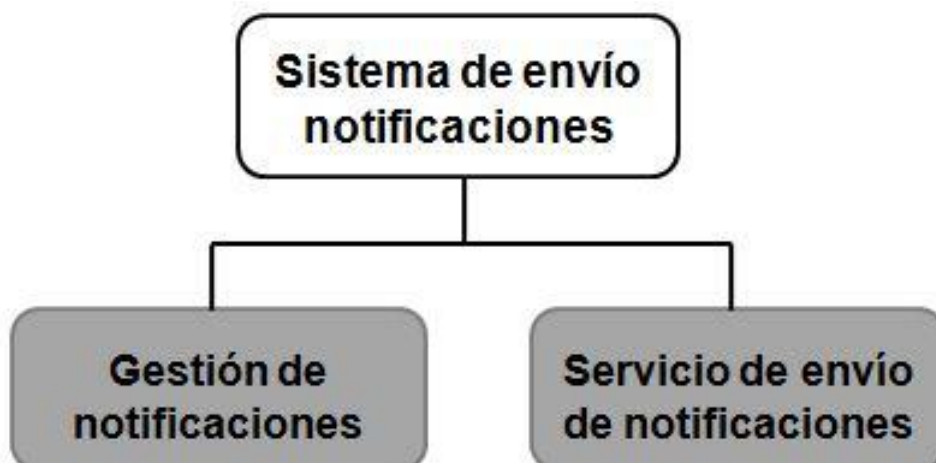


Ilustración 1: Subsistemas del sistema de notificaciones

2.1.1 Gestión de notificaciones

Este subsistema permite a los administradores, configurar, controlar y realizar un seguimiento, sobre todo lo relativo al sistema, permitiendo así gestionar los contactos a los que se les envía las notificaciones, los tipos de notificaciones y las configuraciones correspondientes a cada vía de comunicación. Estos iniciarán el servidor para que los usuarios puedan enviarlas notificaciones y a partir de la información de los diferentes usuarios podrán conocer, por ejemplo: que servicios de notificación son los más utilizados y qué tipo de notificación es la más común.

2.1.2 Servicio de envío de notificaciones

Este subsistema es un daemon¹² que se encarga de gestionar el envío de las notificaciones por cada una de las vías seleccionadas por el usuario.

El proceso de envío de notificaciones se realiza del siguiente modo:

El flujo comienza cuando el cliente GRHS detecta una incidencia y envía la información referente a la misma al servidor de GRHS, este a su vez consume el servicio Rest del sistema de envío de notificaciones. Luego el sistema de envío de notificaciones se encarga de crear la notificación y almacenarla en la base de datos. El subsistema servicio de envío de notificaciones se encuentra constantemente realizándole consultas a la base de datos para detectar si existe una nueva notificación. En este subsistema se encuentran el conjunto de cuatro plugins que se encargan de la gestión de las notificaciones por las diferentes vías de comunicación.

Notificaciones enviadas vía:

Correo electrónico: El plugin de envío de notificaciones por la vía correo se ejecuta como hilo independiente a los otros, está preguntando constantemente a la base de datos si hay una nueva notificación, una vez esto ocurra selecciona la notificación, de esta verifica que la vía correo electrónico fuera seleccionada por el usuario. Una vez hecho esto pasa a tomar de la base de datos la configuración que este activa en el sistema de notificaciones, se autentifica con el servidor pertinente para dicha configuración, luego toma el mensaje de la notificación y el nombre de los contactos, lee de la tabla de la base de datos de contactos la dirección de cada contacto seleccionado y envía el mensaje a cada una de dichas direcciones.

Mensajería Instantánea: El plugin de envío de notificaciones vía mensajería instantánea se ejecuta como hilo independiente a los otros, está preguntando constantemente a la base de datos si hay una nueva notificación, una vez esto ocurra selecciona la notificación, de esta verifica que la vía mensajería instantánea fuera seleccionada por el usuario. Una vez hecho esto pasa a tomar de la base de datos

¹² Daemon: Traducido literalmente significa demonio. Proceso en segundo plano (proceso de ejecución independiente). Aplicación que está permanentemente en estado de alerta en un servidor internet con el fin de realizar determinadas tareas como, por ejemplo, enviar un mensaje de correo electrónico o servir una página web.

la configuración que este activa en el sistema de notificaciones, se autentifica con el servidor pertinente para dicha configuración, luego toma el mensaje de la notificación y el nombre de los contactos, lee de la tabla de la base de datos de contactos la dirección de cada contacto seleccionado y envía el mensaje a cada una de dichas direcciones.

SMS: El plugin de envío de notificaciones por la vía SMS se ejecuta como hilo independiente a los otros, está preguntando constantemente a la base de datos si hay una nueva notificación, una vez esto ocurra selecciona la notificación sin enviar, de esta verifica que la vía SMS fuera seleccionada por el usuario. Una vez hecho esto pasa a tomar de la base de datos la configuración que este activa en el sistema de notificaciones, se autentifica con el servidor pertinente para dicha configuración, luego toma el mensaje de la notificación y el nombre de los contactos, lee de la tabla de la base de datos de contactos la dirección de cada contacto seleccionado y envía el mensaje a cada uno de dichos números telefónicos.

VoIP: El plugin de envío de notificaciones por la vía VoIP se ejecuta como hilo independiente a los otros, está preguntando constantemente a la base de datos si hay una nueva notificación, una vez esto ocurra selecciona la notificación sin enviar, de esta verifica que la vía VoIP fuera seleccionada por el usuario. Una vez hecho esto pasa a tomar de la base de datos la configuración que este activa en el sistema de notificaciones, se autentifica con el servidor pertinente para dicha configuración, luego toma el mensaje de la notificación y el nombre de los contactos, lee de la tabla de la base de datos de contactos la dirección de cada contacto seleccionado y envía el mensaje a cada uno de dichos números telefónicos.

2.2 Lista de funcionalidades

- Gestionar notificación
 - Crear notificación.
 - Listar notificaciones.
 - Eliminar notificación.
 - Detalles de la notificación.
- Gestionar tipo de notificación.
 - Listar tipo de notificación.
 - Adicionar tipo de notificación.
 - Editar tipo de notificación.
 - Eliminar tipo de notificación.
- Gestionar configuración del plugin del correo electrónico.

- Listar configuraciones de correo electrónico.
- Adicionar configuración de correo electrónico.
- Editar configuración de correo electrónico.
- Eliminar configuración de correo electrónico.
- Detalles de la configuración de correo electrónico.
- Gestionar configuración del plugin de mensajería instantánea.
 - Listar configuraciones de mensajería instantánea.
 - Adicionar configuración de mensajería instantánea.
 - Editar configuración de mensajería instantánea.
 - Eliminar configuración de mensajería instantánea.
 - Detalles de la configuración de mensajería instantánea.
- Gestionar configuración del plugin de SMS.
 - Listar configuraciones de SMS.
 - Adicionar configuración de SMS.
 - Editar configuración de SMS.
 - Eliminar configuración de SMS.
 - Detalles de la configuración de SMS.
- Gestionar configuración del plugin de VoIP.
 - Listar configuraciones de VoIP.
 - Adicionar configuración de VoIP.
 - Editar configuración de VoIP.
 - Eliminar configuración de VoIP.
 - Detalles de la configuración de VoIP.
- Gestionar contactos.
 - Listar contactos.
 - Adicionar contacto.
 - Editar contacto.
 - Eliminar contacto.

2.3 Características del sistema.

2.3.1 Usabilidad

Se requiere un nivel medio o alto de conocimientos de informática. Además, el usuario necesitará una preparación previa para operar con el sistema. La aplicación posee un buen diseño e interfaces intuitivas, que permiten una mejor comprensión por el usuario.

2.3.2 Eficiencia

El sistema debe responder en el mínimo de tiempo posible, ante las notificaciones por parte de las aplicaciones clientes. La eficiencia de la aplicación estará determinada en gran medida por la velocidad de las consultas a la base de datos, la rapidez en el envío de las notificaciones, las características de la computadora donde se encuentre instalado el sistema y la velocidad de respuesta de los servidores a los que se conecta la aplicación. El equipo de desarrollo determinó que el tiempo mínimo de respuesta para que la aplicación sea eficiente es de 5 segundos.

2.3.3 Software

Características de software en el servidor:

- Sistema operativo: Debian 7 u 8, Ubuntu 14.04 o Centos 6 o 7.
- Deben ser instaladas las siguientes dependencias:
 - python-setuptools(Repositorio)
 - Django-bootstrap-from-master
 - Requests-1.2.0
 - Django-rest-freamework-master
 - Django-1.7
 - python-psycpg2(Repositorio)
 - python-ldap(Repositorio)

Características de software en el cliente:

- Sistema operativo: Debian, Ubuntu, Nova o Windows XP, 7, 8.
- Deben ser instaladas las siguientes dependencias:
 - setuptools-0.6c9.tar.gz
 - bottle-0.11.66-py2.7.egg
 - pyudev-0.16.1-py2.7.egg
 - python_dmidecode-3.10.13_1.1-py2.7.egg

- lockfile-0.8-py2.7.egg
- logbook-0.4.1-py2.7.egg
- peewee-2.0.4-py2.7.egg
- PyYAML-3.1.10-py2.7.egg
- requests-1.1.0-py2.7.egg
- waitress-0.8.5-py2.7.egg
- Jinja2-2.7.2.egg
- pydi-0.2-py2.7.egg
- python_daemon-1.5.5-py2.7.egg
- heapsort-0.1-py2.7.egg
- python-libxml2 (repositorio)
- gst-0.10-py2.7
- reportlab-3.0.tar.gz

2.3.4 Interfaz

La solución propuesta poseerá una interfaz que cumpla con las pautas de diseño elaboradas por la Universidad de las Ciencias Informáticas, permitiendo que la misma cuente con una interfaz fácil de utilizar para los usuarios que interactúen con ella.(22)

2.4 Fase de Exploración

Es la primera fase de la metodología XP; en esta fase se define el alcance real del proyecto, además se realizan entrevistas con el(los) cliente(s) donde se planean a grandes rasgos las historias de usuario que son de su interés para la entrega del producto, las cuales les posibilitan a los programadores estimar la duración del proyecto. Durante esta fase el equipo de desarrollo se familiariza con las herramientas, lenguajes y aplicaciones necesarias para el desarrollo del sistema.

2.4.1 Historias de usuarios

Las historias de usuarios (HU) son la técnica utilizada en XP para representar los requisitos del software con pequeños textos en los que el cliente detalla una actividad que realizará el sistema de forma sencilla y clara, mostrando solamente el perfil de la tarea a realizarse. Para hacer más comprensible las mismas, a continuación, se describe su leyenda:

- **Número:** número de la historia de usuario incremental en el tiempo.
- **Nombre de la historia de usuario:** nombre de la historia de usuario especificado por el programador.

- **Usuario:** personas involucradas en el desarrollo de las HU.
- **Iteración asignada:** número de la iteración.
- **Prioridad en negocio (Baja, Media, Alta):**
 - Baja: Se le otorga a las HU que son de funcionalidades auxiliares y que son independientes del sistema.
 - Media: Se le otorga a las HU que son de funcionalidades a tener en cuenta, sin que estas tengan una afectación sobre el sistema que se esté desarrollando.
 - Alta: Se le otorga a las HU que son de funcionalidades fundamentales en el desarrollo del sistema.
- **Riesgo en desarrollo (Bajo, Medio, Alto):**
 - Bajo: Cuando en la implementación de las HU puedan existir errores, pero éstos son tratados fácilmente y no afectan el desarrollo del sistema.
 - Medio: Cuando en la implementación de las HU puedan existir errores y retrasen la entrega del producto.
 - Alto: Cuando en la implementación de las HU pueda existir algún error y afecte la disponibilidad del sistema.
- **Puntos estimados:** tiempo estimado que se demorará el desarrollo de la HU.
- **Descripción:** breve descripción de la HU.
- **Observaciones:** Señalamiento o advertencia del sistema.
- **Prototipo de interfaz:** Prototipo de interfaz solo si aplica.(23)

A continuación, se describen tres de las historias de usuario con las que cuenta el sistema:

Tabla 1: HU # 1 Crear notificación.

Historia de Usuario	
Número: 1	Usuario: Usuario
Nombre de historia: Crear notificación.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 3/5	Iteración asignada: 1
Programadores responsables: Elaine de la Caridad Díaz Rodríguez, Víctor Raúl Ramos Linares	

Descripción: El usuario crea la notificación introduciendo los datos: tipo de notificación, contacto(s) para enviar la notificación y el mensaje asociado a la misma. Luego el usuario selecciona por cual o cuales vías realizará el envío de la notificación, las posibles vías a escoger son correo electrónico, mensajería instantánea, SMS y VoIP. Una vez creada la notificación el usuario presiona el botón “Accept” y automáticamente gestiona el envío del mensaje por las vías seleccionadas en dicha notificación. Siempre validando que los datos introducidos por el usuario al crear la notificación, sean los correctos y cumplan con los parámetros establecidos por el programador.

Observaciones: En dependencia de la vía de comunicación seleccionada el proceso de envío de la notificación se realizará de una forma u otra. No pueden existir campos obligatorios en vacíos.

Prototipo de Interfaz:

The image shows a software interface for adding a notification. The window has a title bar with the word "Add" and a close button. The main area contains the following elements:

- Type:***: A dropdown menu with "--Select--" as the selected option.
- Contact(s):***: A list box containing three names: "Victor Raul", "Gerardo Dias", and "Elaine Diaz".
- Message:***: A large, empty text area for entering the notification message.
- Email:** A checkbox that is currently unchecked.
- Instant Messaging:** A checkbox that is currently unchecked.
- SMS:** A checkbox that is currently unchecked.
- VoIP:** A checkbox that is currently unchecked.

At the bottom of the window, there are two buttons: "Accept" (with a checkmark icon) and "Cancel" (with a close icon).

Tabla 2: HU # 2 Listar notificaciones.

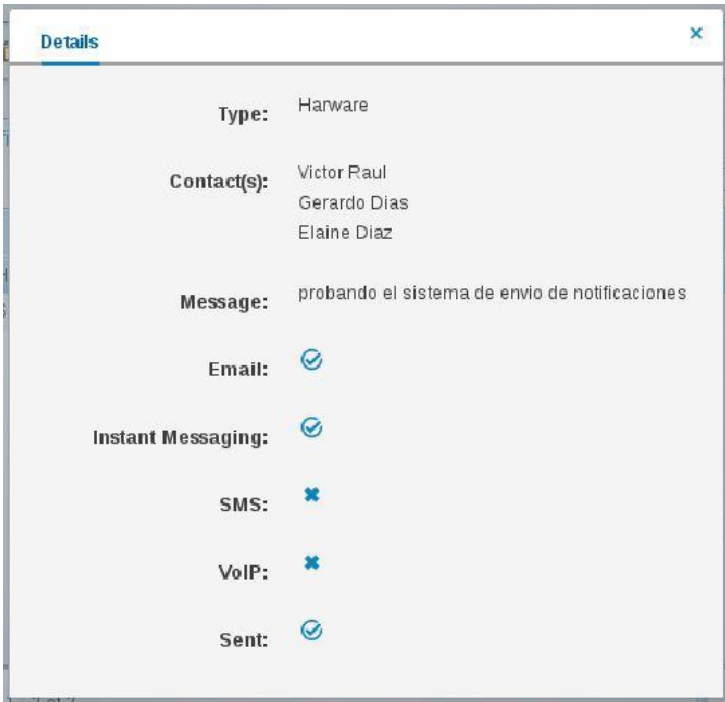
Historia de Usuario																						
Número: 2	Usuario: Usuario																					
Nombre de historia: Listar notificación.																						
Prioridad en negocio: Media	Riesgo en desarrollo: Media																					
Puntos estimados: 1/5	Iteración asignada: 2																					
Programadores responsables: Elaine de la Caridad Díaz Rodríguez, Víctor Raúl Ramos Linares																						
<p>Descripción: El usuario da clic en la opción “Notifications List” y el sistema muestra una lista de todas las notificaciones creadas por el usuario, junto al tipo de notificación, el mensaje asociado a cada una de estas y el estado de la notificación por cada una de las vías (que pueden ser correo electrónico, mensajería instantánea, SMS y VoIP) y el estado que puede ser: enviada o no enviada.</p>																						
Observaciones:																						
<p>Prototipo de Interfaz:</p> <table border="1"> <thead> <tr> <th>Type</th> <th>Message</th> <th>Send Email</th> <th>Send IM</th> <th>Send SMS</th> <th>Send VoIP</th> <th>Sent</th> </tr> </thead> <tbody> <tr> <td>Hardware</td> <td>probando el sisten</td> <td>✓</td> <td>✓</td> <td>✗</td> <td>✗</td> <td>✓</td> </tr> <tr> <td>Software</td> <td>qweqwecv</td> <td>✓</td> <td>✗</td> <td>✗</td> <td>✗</td> <td>✗</td> </tr> </tbody> </table>		Type	Message	Send Email	Send IM	Send SMS	Send VoIP	Sent	Hardware	probando el sisten	✓	✓	✗	✗	✓	Software	qweqwecv	✓	✗	✗	✗	✗
Type	Message	Send Email	Send IM	Send SMS	Send VoIP	Sent																
Hardware	probando el sisten	✓	✓	✗	✗	✓																
Software	qweqwecv	✓	✗	✗	✗	✗																

Tabla 3: HU # 3 Eliminar notificación.

Historia de Usuario	
Número: 3	Usuario: Usuario
Nombre de historia: Eliminar notificación.	
Prioridad en negocio: Baja	Riesgo en desarrollo: Baja
Puntos estimados: 2/5	Iteración asignada: 3
Programadores responsables: Elaine de la Caridad Díaz Rodríguez, Víctor Raúl Ramos Linares	
Descripción: El usuario selecciona una notificación y da clic en la opción “Delete” para eliminarla. Automáticamente el sistema muestra una ventana emergente para confirmar si desea eliminarla. El usuario ratifica la opción de eliminar dando clic en el botón “Accept” y esta se elimina.	
Observaciones:	
Prototipo de Interfaz:	
	

Tabla 4: HU # 4 Detalles de la notificación.

Historia de Usuario	
Número: 4	Usuario: Usuario
Nombre de historia: Detalles de la notificación	

Prioridad en negocio: Baja	Riesgo en desarrollo: Baja
Puntos estimados: 1/5	Iteración asignada: 3
Programadores responsables: Elaine de la Caridad Díaz Rodríguez, Víctor Raúl Ramos Linares	
<p>Descripción: El usuario da clic en la opción “Details” y el sistema muestra una ventana emergente con los datos referentes a la notificación seleccionada, los datos son: tipo de notificación, nombre(s) de lo(s) contacto(s) seleccionados, mensaje y muestra las vías de comunicación correo electrónico, mensajería instantánea, SMS y VoIP, resaltando por cual(es) se envió la notificación y el estado asociado a la misma.</p>	
Observaciones:	
<p>Prototipo de Interfaz:</p> 	

Para consultar el resto de las HU ver [ANEXO 1](#).

2.5 Fase de planificación

La fase de planificación es una de las fases que define la metodología XP, en esta se define el alcance real del sistema permitiendo que el equipo de desarrollo se familiarice con las herramientas, tecnologías y procesos. Durante esta fase el cliente establece la prioridad de cada historia de usuario, y en

consecuencia los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Esta estimación se expresa utilizando como medida el punto. Un punto es considerado como una semana ideal de trabajo que constará de 5 días laborables de 8 horas cada uno para un total de 40 horas a la semana, donde los miembros del equipo de desarrollo trabajan el tiempo planeado sin interrupción. Se toman acuerdos sobre el contenido de la entrega y se determina un cronograma en conjunto con el cliente.

2.5.1 Estimación del esfuerzo de cada historia de usuario

La medida utilizada para la estimación del esfuerzo asociado a la implementación de cada una de las historias de usuario es el punto (máximo esfuerzo), la cual queda establecida por el equipo de desarrollo. Esta medida habitualmente toma valores de 1 a 3 en dependencia de la complejidad de la HU, partiendo de los factores que el equipo decida para clasificarlas. Estos factores por lo general son: la complejidad, el esfuerzo y el riesgo que demande cada una de ellas. Cuando una HU tiene tiempo de duración inferior a una semana significa que está muy bien descrita y que puede ser combinada con otra si muchas dificultades.

Tabla 5: Estimación de esfuerzo por HU.

Número	Historia de Usuario	Puntos estimados
1	Crear notificación.	4/5
2	Listar notificaciones.	2/5
3	Eliminar notificación.	2/5
4	Detalles de la notificación.	1/5
5	Listar tipo de notificación.	1/5
6	Adicionar tipo de notificación.	1/5
7	Editar tipo de notificación.	2/5
8	Eliminar tipo de notificación.	1/5
9	Listar configuraciones de correo electrónico.	2/5
10	Adicionar configuración de correo electrónico.	4/5
11	Editar configuración de correo electrónico.	2/5
12	Eliminar configuración de correo electrónico.	1/5

13	Detalles de la configuración de correo electrónico.	1/5
14	Listar configuraciones de mensajería instantánea.	2/5
15	Adicionar configuración de mensajería instantánea.	4/5
16	Editar configuración de mensajería instantánea.	2/5
17	Eliminar configuración de mensajería instantánea.	1/5
18	Detalles de la configuración de mensajería instantánea.	1/5
19	Listar configuraciones de SMS.	2/5
20	Adicionar configuración de SMS.	4/5
21	Editar configuración de SMS.	2/5
22	Eliminar configuración de SMS.	1/5
23	Detalles de la configuración de SMS.	1/5
24	Listar configuraciones de VoIP.	2/5
25	Adicionar configuración de VoIP.	4/5
26	Editar configuración de VoIP.	2/5
27	Eliminar configuración de VoIP.	2/5
28	Detalles de la configuración de VoIP.	1/5
29	Listar contactos.	2/5
30	Adicionar contacto.	4/5
31	Editar contacto.	2/5
32	Eliminar contacto.	2/5

2.5.2 Plan de iteraciones

Luego de ser identificada, descrita y estimada cada una de las HU se procede a planificar la fase de implementación del proyecto, la cual se realizará en 3 iteraciones. Al finalizar cada una de las

funcionalidades, estas serán mostradas al cliente con el objetivo de detectar cambios necesarios o errores existentes. Las iteraciones estarán conformadas según se explica a continuación:

2.5.2.1 Iteración 1

La primera iteración persigue como objetivo la implementación de las historias de usuario de prioridad alta. Al finalizar esta iteración se realizará la entrega de la primera versión del sistema. Las historias de usuario a implementar en esta iteración son:(1, 6, 10, 15, 20, 25, 30).

2.5.2.2 Iteración 2

El objetivo de esta segunda iteración es la implementación de las historias de usuario de prioridad media. Al finalizar la misma se contará con una versión funcional del sistema, donde se podrán probar las funcionalidades más relevantes del sistema. Las historias de usuario que serán implementadas en esta iteración son: (2, 5, 7, 9, 11, 14, 16, 19, 21, 24, 26, 29, 31).

2.5.2.3 Iteración 3

En la tercera iteración se realizará el desarrollo de las historias de usuario de prioridad baja. Al concluir esta se contará con la versión 1.0 del producto final, adicionando a las funcionalidades resultantes de esta iteración el producto final de la realización de las anteriores iteraciones. Como resultado final de esta iteración se contará con el sistema listo para su despliegue. Las historias de usuario a implementar son: (3, 4, 8, 12, 13, 17, 18, 22, 23, 27, 28, 32).

2.5.3 Plan de duración de iteraciones

XP como parte de su ciclo de vida crea el plan de duración de iteraciones, este plan se encarga de mostrar las historias de usuarios en el orden en que se implementarán en cada una de las iteraciones, así como la duración estimada de cada una de estas.

Tabla 6: Plan de duración de las iteraciones.

Iteración	Orden de la HU a implementar	Duración total
1	Crear notificación. Adicionar tipo de notificación. Adicionar configuración de correo electrónico. Adicionar configuración de mensajería instantánea. Adicionar configuración de SMS. Adicionar configuración de VoIP.	5 semanas

	Adicionar contacto.	
2	<p>Listar notificaciones.</p> <p>Listar tipo de notificación.</p> <p>Editar tipo de notificación.</p> <p>Listar configuraciones de correo electrónico.</p> <p>Editar configuración de correo electrónico.</p> <p>Listar configuraciones de mensajería instantánea.</p> <p>Editar configuración de mensajería instantánea.</p> <p>Listar configuraciones de SMS.</p> <p>Editar configuración de SMS.</p> <p>Listar configuraciones de VoIP.</p> <p>Editar configuración de VoIP.</p> <p>Listar contactos.</p> <p>Editar contacto.</p>	5 semanas
3	<p>Eliminar notificación.</p> <p>Detalles de la notificación.</p> <p>Eliminar tipo de notificación.</p> <p>Eliminar configuración de correo electrónico.</p> <p>Detalles de la configuración de correo electrónico.</p> <p>Eliminar configuración de mensajería instantánea.</p> <p>Detalles de la configuración de mensajería instantánea.</p> <p>Eliminar configuración de SMS.</p>	3 semanas

	<p>Detalles de la configuración de SMS.</p> <p>Eliminar configuración de VoIP.</p> <p>Detalles de la configuración de VoIP.</p> <p>Eliminar contactos.</p>	
--	--	--

2.5.4 Plan de entregas

En este plan se detalla la fecha fin de cada iteración, mostrando una versión desarrollada del producto hasta ese momento, en aras de lograr el producto final en la fecha establecida. A continuación, se muestra el plan de entrega para cada iteración:

Tabla 7: Plan de entregas

Iteración	Fecha final de la iteración
Iteración1	18/03/2016
Iteración 2	22/04/2016
Iteración 3	13/05/2016

2.6 Conclusiones parciales

En este capítulo se analizó la propuesta de solución y se llevaron a cabo las fases de: exploración y planificación, propuestas por la metodología XP, concluyéndose que:

- ✓ La propuesta de solución del sistema a desarrollar, permite que se cumpla con el problema a resolver planteado al comienzo de la presente investigación ya que realiza el envío de las notificaciones mediante todas las vías de comunicación.
- ✓ Se determinaron la lista de funcionalidades con sus historias de usuario asociadas y la lista de reserva del producto, permitiendo definir las iteraciones que se realizarán en aras de organizar el desarrollo del proyecto y elaborar el plan de entregas del mismo.

CAPÍTULO 3 DISEÑO, IMPLEMENTACIÓN Y PRUEBA

En este capítulo se describen las fases de diseño, implementación y prueba, propias de la metodología XP. Se definirá la arquitectura a utilizar en el sistema. Se definen los patrones arquitectónicos y de diseño utilizados en el desarrollo del sistema para su correcta implementación. Se definen las tarjetas Clase-Responsabilidad-Colaborador (CRC) para identificar y organizar las clases orientadas a objetos. Se exponen mediante los diagramas correspondientes las clases persistentes y el modelo físico de la base de datos. Se describirán las tareas de ingeniería generadas para cada historia de usuario. Por último, se realizarán las pruebas al producto para comprobar su correcto funcionamiento y si las funcionalidades se han implementado de forma adecuada.

3.1 Arquitectura de Software

La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente, y los principios que orientan su diseño y evolución.(24)

3.1.1 Arquitectura Cliente-Servidor

En la arquitectura cliente-servidor, el cliente envía un mensaje al servidor solicitando un determinado servicio (petición), y el servidor envía uno o varios mensajes con la respuesta. Mediante esta arquitectura el usuario (cliente) puede acceder a la información sin tener en cuenta su ubicación física y donde pueda estar alojada la misma.

Para el desarrollo del sistema de envío de notificaciones para GRHS se definió la arquitectura cliente-servidor. Se emplea esta arquitectura ya que el sistema GRHS también hace uso de ella.

A continuación, se muestra en la ilustración # 2 la arquitectura cliente-servidor.

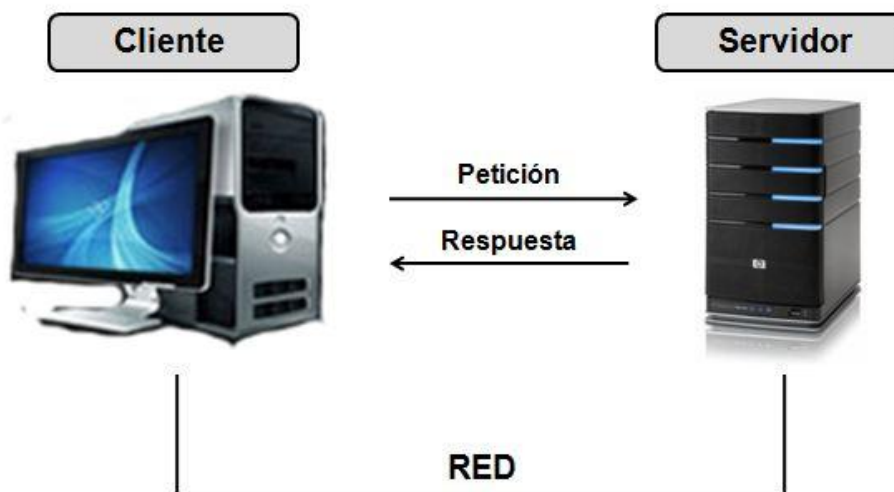


Ilustración 2: Arquitectura Cliente-Servidor.

3.2 Patrón arquitectónico

Los patrones arquitectónicos son los que definen la estructura de un sistema, los cuales a su vez se componen de subsistemas con sus responsabilidades, también tienen una serie de directivas para organizar los componentes del mismo sistema, con el objetivo de facilitar la tarea del diseño de tal sistema(25). Para el desarrollo del sistema el patrón a seguir es Modelo-Vista-Plantilla.

3.2.1 Modelo-Vista-Plantilla

Uno de los frameworks a utilizar para el desarrollo de la solución es Django, el cual usa el Modelo-Vista-Plantilla como una modificación del patrón Modelo-Vista-Controlador (MVC), empleado en la implantación de la solución. En Django el modelo sigue siendo modelo, la vista pasa a hacer la plantilla y el controlador la vista.

- **Modelo:** la capa de acceso a la base de datos. Esta capa contiene toda la información sobre los datos mostrando: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tienen, y las relaciones entre ellos.
- **Plantilla:** la capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación: como algunas cosas son mostradas sobre una página web u otro tipo de documento.
- **Vista:** la capa de la lógica de negocios. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada: es como un puente entre los modelos y las plantillas(25).

Este patrón comienza su funcionamiento cuando el navegador envía una solicitud a la vista. La vista interactúa con el modelo para obtener los datos. La vista llama a la plantilla y esta envía la respuesta de la solicitud al navegador. En la siguiente ilustración se muestra el funcionamiento del patrón MTV:

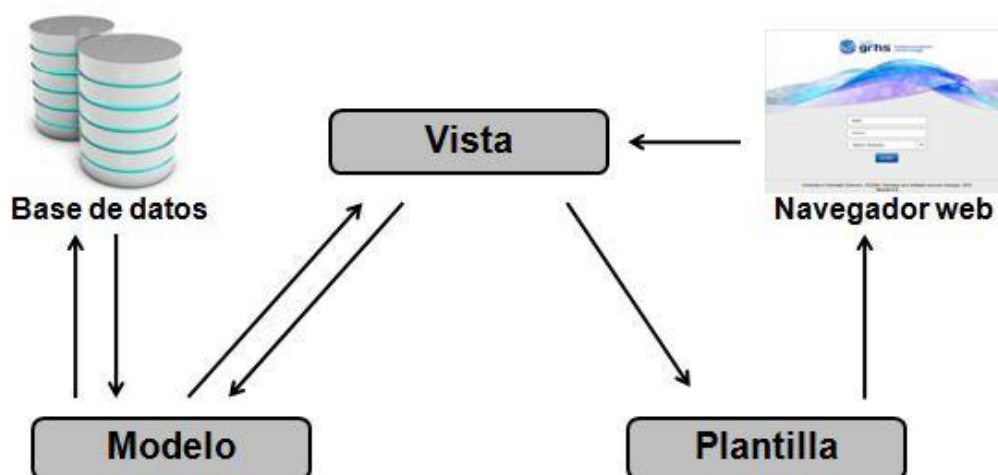


Ilustración 3: Patrón arquitectónico: Modelo –Plantilla –Vista

Para la implementación del sistema de envío de notificaciones se utiliza este patrón, ya que el sistema GRHS también hace uso de este.

3.3 Patrones de diseño

Un patrón de diseño es un conjunto de reglas que describen como afrontar tareas y solucionar problemas que surgen durante el desarrollo del software.

3.3.1 Patrón GRASP

Los patrones GRASP¹³ describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones.(26) A continuación se mencionan y describen los patrones de diseño que se utilizaron durante el desarrollo del sistema:

3.3.1.1 Controlador

Problema: ¿Quién gestiona un evento del sistema? Solución: Asignar la responsabilidad de gestionar un mensaje de un evento del sistema a una clase que represente el sistema global, dispositivo o subsistema (controlador de fachada).(26).

El patrón controlador se ve evidenciado en la clase urls.py, perteneciente al subsistema gestión de notificaciones. Esta clase en dependencia de la petición que llegue se encarga de decir que vista lo va a atender, demostrando que en esta se cumple el patrón controlador.

¹³ GRASP: General Responsibility Assignment Software Patterns, Patrones Generales de Software para Asignación de Responsabilidades.

```

# coding=utf-8
from django.conf.urls import patterns, include
from tlm_web import settings
from django.views.generic.base import RedirectView
from src import url

js_info_dict = {
    'packages': ('src.vendor',
                'src.vendor.apps.accounts'
    ),
}

urlpatterns = patterns(
    '',
    url(r'^login/$', 'src.vendor.views.log_on', name='ingresar'),
    url(r'^$', RedirectView.as_view(url='/web/login')),
    url(r'^il8n/setlang/$', "src.vendor.views.set_language", name="setlan"),
    url(r'^jsil8n/$', 'django.views.il8n.javascript_catalog', js_info_dict,
        name="javascript_catalog"),
    url(r'^accounts/',
        include('src.vendor.apps.accounts.urls',
              namespace='accounts')),
    url(r'^mensajeria/',
        include('src.apps.mensajeria.urls',
              namespace='mensajeria')),
    url(r'^correo/',
        include('src.apps.correo.urls',
              namespace='correo')),
    url(r'^jabber/',
        include('src.apps.jabber.urls',
              namespace='jabber')),
    url(r'^sms/',
        include('src.apps.sms.urls',
              namespace='sms')),
    url(r'^void_ip/',
        include('src.apps.void_ip.urls',
              namespace='void_ip')),

```

Ilustración 4: Clase urls.py

3.3.1.2 Alta cohesión

Problema: ¿Cómo mantener manejable la complejidad? Solución: Asignar responsabilidades de manera que la información que almacena una clase sea coherente y esté relacionada con la clase(26).

El patrón alta cohesión se ve evidenciado en la clase Notificaciones, perteneciente al subsistema gestión de notificaciones. En la siguiente figura se muestra como la clase Notificaciones solo le pide a las restantes clases los datos que para ella son necesarios. Ejemplo: Solo le pide a la clase Contactos el id, aunque esta clase cuenta con más datos.

```

class Notificacion(models.Model):
    tipos = models.ForeignKey(Tipo)
    contactos = models.ManyToManyField(Contacto)
    message = models.CharField(null=True, max_length=148)
    correo = models.BooleanField()
    jabber = models.BooleanField()
    sms = models.BooleanField()
    void_ip = models.BooleanField()
    estado_correo = models.BooleanField()
    estado_jabber = models.BooleanField()
    estado_sms = models.BooleanField()
    estado_voip = models.BooleanField()
    enviada = models.BooleanField()
    revizada = models.BooleanField()

```

Ilustración 5: Clase Notificaciones

3.3.1.3 Bajo acoplamiento

Problema: ¿Cómo dar soporte a las bajas dependencias y al incremento de la reutilización?
Solución: Diseñar con el objetivo de tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases(27).

El patrón bajo acoplamiento se ve evidenciado en la clase CorreoSerializer, perteneciente al subsistema gestión de notificaciones. Esto se demuestra ya que esta clase tiene la mínima dependencia, en este caso la clase CorreoSerializer solo depende de la clase Correo.

```

from rest_framework import serializers
from src.apps.correo.models import *
import base64

class CorreoSerializer(serializers.ModelSerializer):
    class Meta:
        model = Correo

    def save(self):
        h = base64.encodestring(self.object.contrasena)
        self.object.contrasena = h
        return super(CorreoSerializer, self).save()

```

Ilustración 6: Clase CorreoSerializer

3.4 Tarjeta Clase –Responsabilidad –Colaborador (CRC)

La metodología de desarrollo XP como parte de la fase de diseño propone el modelado de las tarjetas Clase-Responsabilidad-Colaborador (CRC), estas constituyen un modelo simple de organizar las clases más relevantes para las funcionalidades del sistema, con el objetivo de desarrollar una representación organizada de las clases. A continuación se muestran dos tarjetas CRC, correspondientes a tres de las clases que conforman el sistema desarrollado, el resto se encuentran en el [ANEXO 2](#).

Tabla 8: CRC. Clase: Correo.

Clase: Correo	
Responsabilidad	Colaborador
Esta clase es la encargada de definir el modelo correspondiente a la configuración del correo en la base de datos.	CorreoSerializer

Tabla 9: CRC. Clase: CorreoSerializer.

Clase: CorreoSerializer	
Responsabilidad	Colaborador
Esta clase es la encargada de generar una representación apta para almacenamiento de los datos de la configuración del correo.	Correo

Tabla 10: CRC. Clase: CorreoMainMenu.

Clase: CorreoMainMenu	
Responsabilidad	Colaborador
Esta clase es la encargada de definir la raíz visual de menú correo en la aplicación web.	CorreoMenu ConfiguracionMainMenu

Tabla 11 CRC. Clase: CorreoMenu.

Clase: CorreoMenu

Responsabilidad	Colaborador
Esta clase es la encargada de gestionar todas las funcionalidades correspondientes a la gestión de las notificaciones vía correo electrónico, mostrando la interfaz correspondientes a estas funcionalidades.	CorreoMainMenu

3.5 Diagrama de clases persistentes

El diagrama de clases persistentes muestra las clases que tienen durabilidad en el tiempo y la relación que existe con las restantes clases. A continuación en la ilustración # 7 se muestra el diagrama de clases persistentes referente al sistema de envío de notificaciones vía correo electrónico, mensajería instantánea, SMS y VoIP para el sistema GRHS.

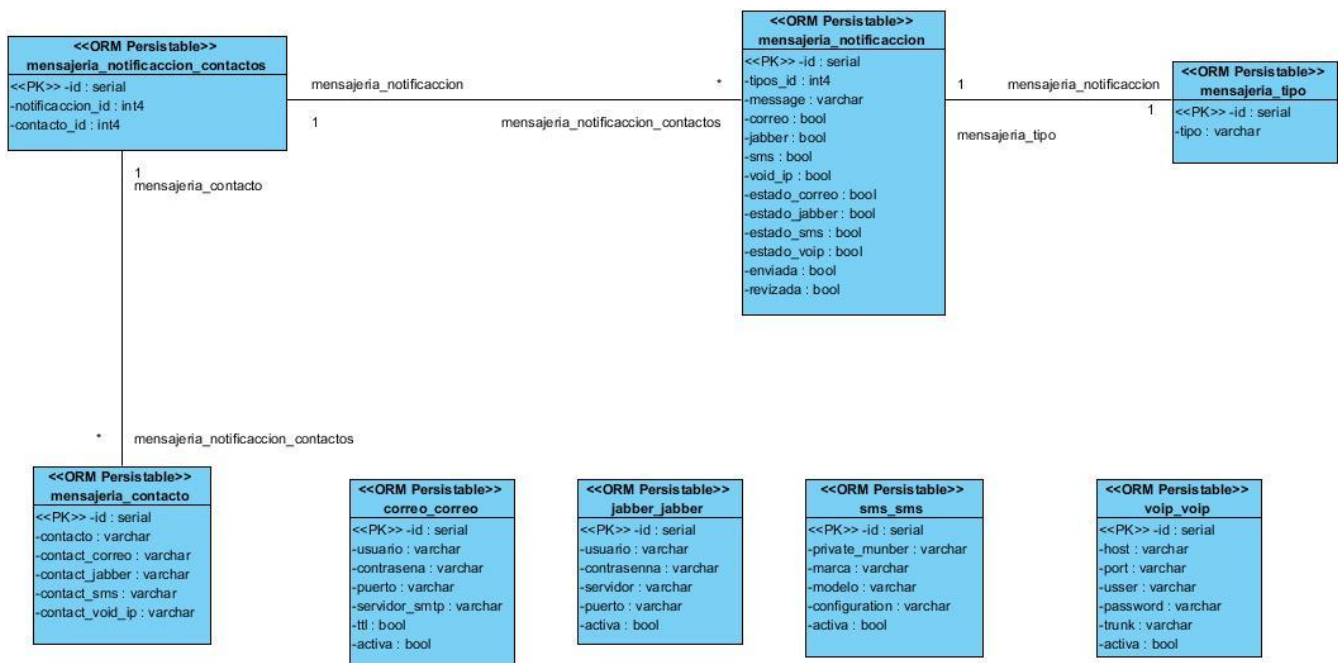


Ilustración 7: Diagrama de clases persistentes

3.6 Modelo físico de la Base de Datos

A continuación en la ilustración # 8 se muestra el modelo físico de la base de datos obtenido del diagrama de clases persistentes, este modelo optimiza el rendimiento a la vez que asegura la integridad de los datos al evitar repeticiones innecesarias de datos, mostrando la relación que existe entre las entidades de la base de datos.(27)



Ilustración 8: Modelo físico de la base de datos

3.7 Tareas de Ingeniería

Las tareas de ingeniería son escritas por los desarrolladores, a partir de las historias de usuario creadas por el cliente, las cuales brindan una mayor información para la implementación. Las tareas de ingeniería se representan mediante tablas, divididas por las siguientes secciones.

- **Número tarea:** los números deben ser consecutivos.
- **Número historia de usuario:** número de la historia de usuario a la que pertenece la tarea.
- **Nombre tarea:** nombre que identifica a la tarea.
- **Tipo de tarea:** las tareas pueden ser de: Desarrollo, Corrección, Mejora, Otra (Especificar)
- **Puntos estimados:** tiempo estimado en semanas que se le asignará a su desarrollo.

- **Fecha inicio:** fecha en que inicia el desarrollo de la tarea.
- **Fecha fin:** fecha en que finaliza el desarrollo de la tarea.
- **Programador responsable:** nombre y apellidos del programador.
- **Descripción:** breve descripción de la tarea.

A continuación, se describen 3 de Tareas de Ingeniería:

Tabla 12: Tarea de Ingeniería # 1 Crear notificación.

Tarea de Ingeniería	
Número tarea: 1	Número historia de usuario: 1
Nombre tarea: Crear notificación	
Tipo de tarea: Desarrollo	Puntos estimados: 3/5
Fecha inicio: 15/2/2016	Fecha fin: 17/2//2016
Programador Responsable: Elaine de la Caridad Díaz Rodríguez, Víctor Raúl Ramos Linares	
<p>Descripción: El usuario da clic en la opción “Add” y el sistema muestra una ventana emergente con los datos: tipo de notificación, contacto(s) para enviar la notificación y el mensaje asociado a la misma. Luego el usuario selecciona por cual o cuales vías realizará el envío de la notificación, las posibles vías a escoger son correo electrónico, mensajería instantánea, SMS y VoIP. Una vez creada la notificación el usuario presiona el botón “Accept” y automáticamente se gestiona el envío del mensaje por las vías seleccionadas en dicha notificación. La aplicación cliente le envía al sistema la notificación que desea enviar.</p>	

Tabla 13: Tarea de Ingeniería # 2 Validar notificación.

Tarea de Ingeniería	
Número tarea: 2	Número historia de usuario: 1
Nombre tarea: Validar notificación	
Tipo de tarea: Desarrollo	Puntos estimados: 1/5
Fecha inicio: 18/2/2016	Fecha fin: 18/2/2016

Programador Responsable: Elaine de la Caridad Díaz Rodríguez, Víctor Raúl Ramos Linares

Descripción: Al ser creada la notificación, el sistema se comunica con el servidor del servicio para verificar que los datos introducidos por el cliente al crear la notificación sean válidos, y cumplan con los parámetros establecidos por el programador. Si esto no se cumple el sistema le muestra al usuario un mensaje de alerta donde muestra que datos no son válidos.

Tabla 14: Tarea de Ingeniería # 3 Listar notificaciones.

Tarea de Ingeniería	
Número tarea: 3	Número historia de usuario: 2
Nombre tarea: Listar notificaciones.	
Tipo de tarea: Desarrollo	Puntos estimados: 2/5
Fecha inicio: 21/3/2016	Fecha fin: 22/3/2016
Programador Responsable: Elaine de la Caridad Díaz Rodríguez, Víctor Raúl Ramos Linares	
Descripción: El usuario da clic en la opción "Notifications List" y el sistema consulta a la base de datos para obtener todas las notificaciones enviadas por un usuario. Luego se muestra una lista de todas las notificaciones creadas por el usuario, junto al tipo de notificación, el mensaje asociado a cada una de estas y el estado de la notificación por cada una de las vías (que pueden ser correo electrónico, mensajería instantánea, SMS y VoIP) y el estado general que puede ser: enviada o no enviada.	

Tabla 15: Tarea de Ingeniería # 4 Eliminar notificación.

Tarea de Ingeniería	
Número tarea: 4	Número historia de usuario: 3
Nombre tarea: Eliminar notificación.	
Tipo de tarea: Desarrollo	Puntos estimados: 2/5
Fecha inicio: 25/4/2016	Fecha fin: 26/4/2016
Programador Responsable: Elaine de la Caridad Díaz Rodríguez, Víctor Raúl Ramos Linares	

Descripción: El sistema muestra la lista con todas las notificaciones existentes en el sistema. El usuario selecciona una notificación de la lista y da clic en la opción “Delete”, automáticamente aparece una ventana emergente para confirmar que se desea eliminar la notificación seleccionada. El usuario ratifica la opción de eliminar dando clic en el botón “Accept”, y el sistema pasa a eliminar la notificación de la base de datos. Luego de eliminada la notificación seleccionada por el usuario el sistema actualiza la lista de notificaciones.

Para consultar el resto de las Tareas de Ingeniería ver [ANEXO 3](#).

3.8 Pruebas

Las pruebas tienen un valor primordial en el desarrollo de aplicaciones ya que un fallo en estas representaría costos elevados. Unos de los pilares de la metodología XP es el proceso de pruebas. XP propone comprobar constantemente cada una de las funcionalidades implementadas. Esto permite aumentar la calidad de los sistemas, reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. La metodología XP divide las pruebas en dos grupos: pruebas unitarias y pruebas de aceptación o funcionales.

3.8.1 Pruebas unitarias

Las pruebas unitarias son una de las piedras angulares de la programación extrema (XP). En XP los programadores deben escribir las pruebas unitarias para cada módulo antes de escribir el código. No es necesario escribir casos de prueba para todos los módulos, sólo para aquellos en que exista la posibilidad de que puedan fallar.(28)

Las pruebas unitarias se le realizaron al sistema a través de la clase TestCase de la librería test del framework Django. Los resultados de las mismas fueron satisfactorios como se muestra en la ilustración # 9.

```
Ran 419 tests in 24.511s
OK (skipped=1)
Destroying test database for alias 'default'...
root@Victor:/home/victor/Desktop/Actual/tlm_web# █
```

Ilustración 9: Prueba unitaria al sistema de envío de notificaciones.

3.8.2 Pruebas de aceptación

Las pruebas de aceptación son especificadas por el cliente y son creadas en base a las historias de usuario. Estas son las encargadas de comprobar que las funcionalidades desarrolladas cumplan con las expectativas del cliente. Estas pruebas indican el camino a seguir en cada iteración, mostrándole al equipo de desarrollo qué camino seguir y en que funcionalidades debe poner mayor atención.

La importancia de estas pruebas se debe a que permiten medir el nivel de satisfacción del cliente al finalizar cada iteración, marcando así el fin de esta y el comienzo de la siguiente. Por ello se muestra a continuación una serie de casos de prueba, los cuales servirán como muestra visual del proceso de pruebas realizadas a la aplicación propuesta. Dichos casos de prueba serán representados mediante tablas divididas por las siguientes secciones:

- **Clases Válidas:** describe los pasos seguidos durante el desarrollo de la prueba, se tiene en cuenta las entradas válidas y opciones seleccionadas por el usuario con el objetivo de verificar si se obtiene el resultado esperado.
- **Clases Inválidas:** describe los pasos seguidos durante el desarrollo de la prueba, se tiene en cuenta las entradas inválidas y opciones seleccionadas por el usuario con el objetivo de verificar si se obtiene el resultado esperado y cómo responde el sistema.
- **Resultado Esperado:** breve descripción del resultado que se espera ya sea para clases válidas o inválidas.
- **Resultado de la Prueba:**
 - **Satisfactorio:** Cuando el resultado de la prueba es exactamente el esperado por el usuario.
 - **No satisfactorio:** Cuando el resultado de la prueba realizada genera un error de codificación en la aplicación o muestra como resultado elementos no deseados o fuera de contexto, trayendo como consecuencia que la funcionalidad requerida por el cliente no tenga resultado, lo que invalida también la UH.
- **Observaciones:** algún señalamiento o advertencia que sea necesario hacerle a la sección que se está probando.

A continuación, se muestra un caso de prueba correspondiente a cinco de las funcionalidades del sistema de envío de notificaciones.

Tabla 16: Caso de Prueba Eliminar notificación.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
El usuario da clic en la opción "Delete" para eliminar una(s) notificación(es) y el sistema muestra una		El sistema muestra una ventana emergente con un mensaje de confirmación "Are you	Satisfactorio.	

ventana emergente con un mensaje de confirmación "Are you sure you want to permanently Delete (the) selected item(s)?" Si el usuario selecciona el botón "Accept", el sistema elimina la(s) notificación(es).		sure you want to permanently Delete (the) selected item(s)?" Si el usuario selecciona el botón "Accept", el sistema elimina la(s) notificación(es).		
---	--	---	--	--

Tabla 17: Caso de Prueba Adicionar configuración de correo electrónico.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
El usuario da clic en la opción "add" dentro del menú "Mail" y el sistema muestra una ventana emergente con los campos: "User", "Password", "Server SMTP" y "Port"; además permite seleccionar Ttl (conexión segura) y la opción "Active" (estado de la configuración). Luego de introducidos todos los datos el usuario da clic en la opción "Accept".		El sistema verifica que ninguno de los campos obligatorios se encuentre en blanco y que la dirección de correo sea correcta y luego adiciona la nueva configuración.	Satisfactorio	

	<p>El usuario da clic en la opción “add” dentro del menú “Mail” y el sistema muestra una ventana emergente con los campos: “User”, “Password”, “Server SMTP” y “Port”; además permite seleccionar Ttl (conexión segura) y la opción “Active” (estado de la configuración).El usuario deja de introducir algún(os) dato(s) o introduce una dirección de correo errónea y da clic en la opción “Accept”.</p>	<p>El sistema verifica que ninguno de los campos obligatorios se encuentre en blanco y que la dirección de correo sea correcta, si alguna de estas variantes no se cumple el sistema le informa al usuario, y la configuración no es adicionada.</p>	<p>Satisfactorio.</p>	
--	--	--	-----------------------	--

Tabla 18: Caso de Prueba Adicionar configuración de mensajería instantánea.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
<p>El usuario da clic en la opción “add” dentro del menú “Instant Messaging” y el sistema muestra una ventana emergente con los campos: “User”, “Password”,</p>		<p>El sistema verifica que los campos “User”, “Password”, “Server” y “Port” no se encuentren vacíos, de ser así adiciona la nueva configuración.</p>	<p>Satisfactorio</p>	

<p>“Server”, “Port” y permite seleccionar la opción “Active” (estado de la configuración). Luego de introducidos todos los datos el usuario da clic en la opción “Accept”.</p>				
	<p>El usuario da clic en la opción “add” dentro del menú “Instant Messaging” y el sistema muestra una ventana emergente con los campos: “User”, “Password”, “Server”, “Port” y permite seleccionar la opción “Active” (estado de la configuración). El usuario deja de introducir algún(os) dato(s) y da clic en la opción “Accept”.</p>	<p>El sistema verifica que los campos “User”, “Password”, “Server”, “Port” no se encuentren vacíos, al verificar encuentra un campo vacío, por lo que se lo notifica al usuario y la configuración no es adicionada.</p>	<p>Satisfactorio.</p>	

Tabla 19: Caso de Prueba Adicionar configuración de SMS.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
<p>El usuario da clic en la opción “add”</p>		<p>El sistema verifica que los campos “Private</p>	<p>Satisfactorio.</p>	

<p>dentro del menú “SMS” y el sistema muestra una ventana emergente con los campos: “Private number”, “Trademark”, “Model”, “Configuration”; además permite seleccionar la opción “Active” (estado de la configuración). Luego de introducidos todos los datos el usuario da clic en la opción “Accept”.</p>		<p>number”, “Trademark”, “Model” y “Configuration” no se encuentren vacíos, de ser así adiciona la nueva configuración.</p>		
	<p>El usuario da clic en la opción “add” dentro del menú “Instant Messaging” y el sistema muestra una ventana emergente con los campos: “Private number”, “Trademark”, “Model”, “Configuration” y permite seleccionar la opción “Active” (estado de la configuración). El usuario deja de introducir algún(os)</p>	<p>El sistema verifica que los campos “Private number”, “Trademark”, “Model” y “Configuration” no se encuentren vacíos, al verificar encuentra un campo vacío, por lo que se lo notifica al usuario y la configuración no es adicionada.</p>	<p>Satisfactorio.</p>	

	dato(s) y da clic en la opción "Accept".			
--	--	--	--	--

Tabla 20: Caso de Prueba Adicionar configuración de VoIP.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
<p>El usuario da clic en la opción "add" dentro del menú "VoIP" y el sistema muestra una ventana emergente con los campos: "Host", "Port", "User", "Password" y "Trunk" (enlace troncal); además permite seleccionar la opción "Active" (estado de la configuración).</p> <p>Luego de introducidos todos los datos el usuario da clic en la opción "Accept".</p>		<p>El sistema verifica que los campos "Host", "Port", "User", "Password" y "Trunk" (enlace troncal); no se encuentren vacíos, de ser así adiciona la nueva configuración.</p>	Satisfactorio.	
	<p>El usuario da clic en la opción "add" dentro del menú "VoIP" y el sistema muestra una ventana emergente con los campos: "Host", "Port", "User", "Password" y "Trunk" (enlace troncal);</p>	<p>El sistema verifica que los campos "User", "Password" y "Trunk" (enlace troncal) no se encuentren vacíos, al verificar encuentra un campo vacío, el sistema se lo notifica al usuario y la</p>	Satisfactorio.	

	además permite seleccionar la opción "Active" (estado de la configuración). El usuario deja de introducir algún(os) dato(s) y da clic en la opción "Accept".	configuración no es adicionada.		
--	--	---------------------------------	--	--

Para consultar el resto de los casos de prueba consultar ANEXO 4.

Al terminar las pruebas de aceptación se obtuvo un total de 10 no conformidades en la primera iteración, de las cuales 6 fueron resueltas y 4 quedaron pendientes. En la segunda iteración se obtuvieron 6 no conformidades donde 4 fueron resueltas y 2 quedaron pendientes; y en la tercera iteración se encontraron 4 no conformidades y todas fueron resueltas. Los resultados correspondientes a estas pruebas se muestran a continuación:

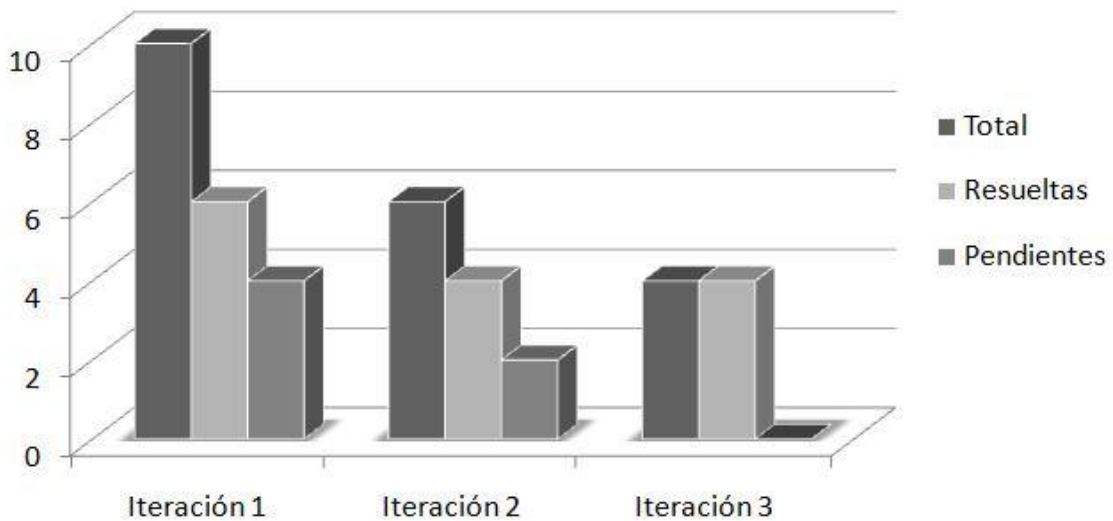


Ilustración 10: Resultados de las pruebas de aceptación.

3.9 Conclusiones parciales

En este capítulo se llevaron a cabo las fases de: diseño, implementación y pruebas, propuestas por la metodología XP, concluyéndose que:

- ✓ La arquitectura cliente servidor es la más adecuada para el desarrollo del sistema de envío de notificaciones.
- ✓ El patrón arquitectónico más apropiado a utilizar para la implementación del sistema es modelo-vista-plantilla debido a que se utiliza Django como uno de los framework de desarrollo.

- ✓ Dentro de los patrones de diseño existentes se utilizan los GRASP debido a que son los que más se ajustan a la solución creada.
- ✓ Se definieron las tarjetas CRC permitiendo identificar y organizar las clases orientadas a objetos.
- ✓ Se describieron las tareas de ingeniería para cada historia de usuario, necesarias para la implementación del sistema.

CONCLUSIONES GENERALES

- ✓ El estudio realizado de las herramientas informáticas existentes arrojó que no es posible el uso de estas en la universidad ya que no cumplen con todos los requerimientos del sistema que se pretende implementar.
- ✓ La utilización de la metodología ágil XP es adecuada para el desarrollo del software, ya que se adapta a las necesidades del mismo por las características del sistema a implementar, permitiendo guiar el proceso de desarrollo del software de forma correcta.
- ✓ Las herramientas y tecnologías fueron escogidas adecuadamente para desarrollar la aplicación, teniendo en cuenta las peculiaridades de cada una de estas y las características del sistema a implementar.
- ✓ Se realizaron pruebas de aceptación y unitarias con el objetivo de comprobar la correcta implementación de cada una de las funcionalidades y así evidenciar la calidad del sistema desarrollado.

RECOMENDACIONES

- ✓ Incluir el sistema resultante de esta investigación en las nuevas versiones del sistema GRHS.
- ✓ Probar el sistema de envío de notificaciones en ambiente real.

REFERENCIAS BIBLIOGRÁFICAS

1. Diccionario de la lengua española - Edición del Tricentenario. *Diccionario de la lengua española* [online]. [Accessed 2 November 2015]. Available from: <http://dle.rae.es/?w=notificaciones&m=form&o=h>
2. RODRÍGUEZ CRESPO, Claudia, RODRÍGUEZ GARCÍA, Odaimys, AZÁN BASALLO, Yasser and RODRÍGUEZ FERNÁNDEZ, Gerardo. \iSistema de integración de servicios de comunicación a usuarios (SISCU) [online]. 2011. [Accessed 27 October 2015]. Available from: http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_04068_11
3. mensajería instantánea | Real Academia de Ingeniería. [online]. [Accessed 27 October 2015]. Available from: <http://diccionario.raing.es/es/lema/mensajer%C3%ADa-instant%C3%A1nea>
4. RODRÍGUEZ CRESPO, Claudia, RODRÍGUEZ GARCÍA, Odaimys, AZÁN BASALLO, Yasser and RODRÍGUEZ FERNÁNDEZ, Gerardo. *Sistema de integración de servicios de comunicación a usuarios (SISCU)* [online]. 2011. [Accessed 27 October 2015]. Available from: http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_04068_11
5. *Microsoft Word - seguridad VOIP FINAL 1.doc - 896.pdf* [online]. [Accessed 27 October 2015]. Available from: <http://www.it-docs.net/ddata/896.pdf>
6. Buscar en el diccionario informático. [online]. [Accessed 12 November 2015]. Available from: <http://www.lawebdelprogramador.com/diccionario/buscar.php?opc=1&charSearch=pluginBuscar> en el diccionario informático - plugin
7. Como tener tus notificaciones de Facebook en el escritorio. *WebAdictos* [online]. [Accessed 2 November 2015]. Available from: <http://webadictos.com/2011/01/20/como-tener-tus-notificaciones-de-facebook-en-el-escritorio/>
8. Servicio de ayuda. [online]. [Accessed 2 November 2015]. Available from: <https://www.facebook.com/help/1668906000006551>
9. MySpace Notifier (Windows). *Uptodown.com* [online]. [Accessed 27 October 2015]. Available from: <http://myspace-notifier.uptodown.com>
10. *alertfind-enterprise-notification-datasheet-27666.pdf* [online]. [Accessed 9 November 2015]. Available from: <http://www.messageone.com/wp-content/uploads/2015/04/alertfind-enterprise-notification-datasheet-27666.pdf>
11. FIGUEROA, Roberth G. and CAMILO J. SOLÍS. Metodologías tradicionales vs. metodologías ágiles. .
12. Welcome to Python.org. *Python.org* [online]. [Accessed 10 December 2015]. Available from: [https://www.python.org/The official home of the Python Programming Language](https://www.python.org/The%20official%20home%20of%20the%20Python%20Programming%20Language)
13. Introducción a JavaScript. [online]. [Accessed 23 April 2016]. Available from: <http://librosweb.es/libro/javascript/>
14. Nuevo Django 1.4. *DesarrolloWeb.com* [online]. [Accessed 2 November 2015]. Available from: [/actualidad/nuevo-django-1-4-6755.html](http://actualidad/nuevo-django-1-4-6755.html)
15. ¿Qué es jQuery? [online]. [Accessed 17 December 2015]. Available from: <http://www.mexired.com/blog/que-es-jquery>

16. Conocimiento Abierto » Blog Archive » Qué es Twitter Bootstrap y cómo aprender a usarlo. [online]. [Accessed 17 December 2015]. Available from: <http://conocimientoabierto.es/que-es-twitter-bootstrap-y-como-aprender-a-usarlo/657/>
17. Backbone.js. [online]. [Accessed 23 April 2016]. Available from: <http://backbonejs.org/>
18. PyCharm el IDE de Python lanza su versión 3.4. *Hipertextual* [online]. 10 June 2014. [Accessed 9 June 2016]. Available from: <http://hipertextual.com/archivo/2014/06/pycharm-ide-python>
19. Herramientas Case. [online]. 11:12:14 UTC. [Accessed 31 March 2016]. Available from: <http://es.slideshare.net/guestf131a9/herramientas-case>
20. Visual Paradigm 8.0 (formerly VP-UML 8.0) Released. [online]. [Accessed 23 April 2016]. Available from: <https://www.visual-paradigm.com/aboutus/newsreleases/vpuml80.jsp>
21. Sobre PostgreSQL | www.postgresql.org.es. [online]. [Accessed 5 April 2016]. Available from: http://www.postgresql.org.es/sobre_postgresql
22. Estrategia Marcaria | UCI. [online]. [Accessed 5 April 2016]. Available from: <http://iux.prod.uci.cu/>
23. DÍAZ ODUARDO, Dianet, GARCÍA NÚÑEZ, Alejandro, PÉREZ CASTILLO, Erick and SEGURA PEÑA, Rainer. *SISTEMA INTEGRADO PARA EL ENVÍO DE NOTIFICACIONES VÍA CORREO ELECTRÓNICO, MENSAJERÍA INSTANTÁNEA, SMS Y LLAMADAS VOIP* [online]. 2014. [Accessed 27 October 2015]. Available from: http://repositorio_institucional.uci.cu//jspui/handle/ident/8222
24. VALDIVIA, Enrique. *Arquitectura de Software: Glosario. Arquitectura de Software* [online]. 23 August 2010. [Accessed 17 February 2016]. Available from: <http://arquisoftware.blogspot.com/2010/08/glosario.html>
25. isg3 / Patrones Arquitectónicos. [online]. [Accessed 17 February 2016]. Available from: <http://isg3.pbworks.com/w/page/7624479/Patrones%20Arquitect%C3%B3nicos>
26. ANDRÉS GROSSO. Patrones GRASP | *Prácticas de Software*. [online]. [Accessed 19 April 2016]. Available from: <http://www.practicadesoftware.com.ar/2011/03/patrones-grasp>
27. IBM Knowledge Center - DB2 10 - Introducción - Diseño físico de base de datos. [online]. [Accessed 17 June 2016]. Available from: http://www.ibm.com/support/knowledgecenter/es/SSEPEK_10.0.0/com.ibm.db2z10.doc.intro/src/tpc/db2z_physicaldatabasesdesign.html
28. Unit Tests. [online]. [Accessed 19 May 2016]. Available from: <http://www.extremeprogramming.org/rules/unittests.html>

BIBLIOGRAFIA

1. Diccionario de la lengua española - Edición del Tricentenario. *Diccionario de la lengua española* [online]. [Accessed 2 November 2015]. Available from: <http://dle.rae.es/?w=notificaciones&m=form&o=h>
2. RODRÍGUEZ CRESPO, Claudia, RODRÍGUEZ GARCÍA, Odaimys, AZÁN BASALLO, Yasser and RODRÍGUEZ FERNÁNDEZ, Gerardo. \iSistema de integración de servicios de comunicación a usuarios (SISCU) [online]. 2011. [Accessed 27 October 2015]. Available from: http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_04068_11
3. mensajería instantánea | Real Academia de Ingeniería. [online]. [Accessed 27 October 2015]. Available from: <http://diccionario.raing.es/es/lema/mensajer%C3%ADa-instant%C3%A1nea>
4. RODRÍGUEZ CRESPO, Claudia, RODRÍGUEZ GARCÍA, Odaimys, AZÁN BASALLO, Yasser and RODRÍGUEZ FERNÁNDEZ, Gerardo. *Sistema de integración de servicios de comunicación a usuarios (SISCU)* [online]. 2011. [Accessed 27 October 2015]. Available from: http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_04068_11
5. *Microsoft Word - seguridad VOIP FINAL 1.doc - 896.pdf* [online]. [Accessed 27 October 2015]. Available from: <http://www.it-docs.net/ddata/896.pdf>
6. Buscar en el diccionario informático. [online]. [Accessed 12 November 2015]. Available from: <http://www.lawebdelprogramador.com/diccionario/buscar.php?opc=1&charSearch=pluginBuscar> en el diccionario informático - plugin
7. Como tener tus notificaciones de Facebook en el escritorio. *WebAdictos* [online]. [Accessed 2 November 2015]. Available from: <http://webadictos.com/2011/01/20/como-tener-tus-notificaciones-de-facebook-en-el-escritorio/>
8. Servicio de ayuda. [online]. [Accessed 2 November 2015]. Available from: <https://www.facebook.com/help/1668906000006551>
9. MySpace Notifier (Windows). *Uptodown.com* [online]. [Accessed 27 October 2015]. Available from: <http://myspace-notifier.uptodown.com>
10. *alertfind-enterprise-notification-datasheet-27666.pdf* [online]. [Accessed 9 November 2015]. Available from: <http://www.messageone.com/wp-content/uploads/2015/04/alertfind-enterprise-notification-datasheet-27666.pdf>
11. FIGUEROA, Roberth G. and CAMILO J. SOLÍS. Metodologías tradicionales vs. metodologías ágiles. .
12. Welcome to Python.org. *Python.org* [online]. [Accessed 10 December 2015]. Available from: [https://www.python.org/The official home of the Python Programming Language](https://www.python.org/The%20official%20home%20of%20the%20Python%20Programming%20Language)
13. Introducción a JavaScript. [online]. [Accessed 23 April 2016]. Available from: <http://librosweb.es/libro/javascript/>
14. Nuevo Django 1.4. *DesarrolloWeb.com* [online]. [Accessed 2 November 2015]. Available from: [/actualidad/nuevo-django-1-4-6755.html](http://actualidad/nuevo-django-1-4-6755.html)
15. ¿Qué es jQuery? [online]. [Accessed 17 December 2015]. Available from: <http://www.mexired.com/blog/que-es-jquery>

16. Conocimiento Abierto » Blog Archive » Qué es Twitter Bootstrap y cómo aprender a usarlo. [online]. [Accessed 17 December 2015]. Available from: <http://conocimientoabierto.es/que-es-twitter-bootstrap-y-como-aprender-a-usarlo/657/>
17. Backbone.js. [online]. [Accessed 23 April 2016]. Available from: <http://backbonejs.org/>
18. PyCharm el IDE de Python lanza su versión 3.4. *Hipertextual* [online]. 10 June 2014. [Accessed 9 June 2016]. Available from: <http://hipertextual.com/archivo/2014/06/pycharm-ide-python/>
19. Herramientas Case. [online]. 11:12:14 UTC. [Accessed 31 March 2016]. Available from: <http://es.slideshare.net/guestf131a9/herramientas-case>
20. Visual Paradigm 8.0 (formerly VP-UML 8.0) Released. [online]. [Accessed 23 April 2016]. Available from: <https://www.visual-paradigm.com/aboutus/newsreleases/vpuml80.jsp>
21. Sobre PostgreSQL | www.postgresql.org.es. [online]. [Accessed 5 April 2016]. Available from: http://www.postgresql.org.es/sobre_postgresql
22. Estrategia Marcaria | UCI. [online]. [Accessed 5 April 2016]. Available from: <http://iux.prod.uci.cu/>
23. DÍAZ ODUARDO, Dianet, GARCÍA NÚÑEZ, Alejandro, PÉREZ CASTILLO, Erick and SEGURA PEÑA, Rainer. *SISTEMA INTEGRADO PARA EL ENVÍO DE NOTIFICACIONES VÍA CORREO ELECTRÓNICO, MENSAJERÍA INSTANTÁNEA, SMS Y LLAMADAS VOIP* [online]. 2014. [Accessed 27 October 2015]. Available from: http://repositorio_institucional.uci.cu/jspui/handle/ident/8222
24. VALDIVIA, Enrique. *Arquitectura de Software: Glosario. Arquitectura de Software* [online]. 23 August 2010. [Accessed 17 February 2016]. Available from: <http://arquisoftware.blogspot.com/2010/08/glosario.html>
25. isg3 / Patrones Arquitectónicos. [online]. [Accessed 17 February 2016]. Available from: <http://isg3.pbworks.com/w/page/7624479/Patrones%20Arquitect%C3%B3nicos>
26. ANDRÉS GROSSO. Patrones GRASP | Prácticas de Software. [online]. [Accessed 19 April 2016]. Available from: <http://www.practicadesoftware.com.ar/2011/03/patrones-grasp>
27. IBM Knowledge Center - DB2 10 - Introducción - Diseño físico de base de datos. [online]. [Accessed 17 June 2016]. Available from: http://www.ibm.com/support/knowledgecenter/es/SSEPEK_10.0.0/com.ibm.db2z10.doc.intro/src/tpc/db2z_physicaldatabasesdesign.html
28. Unit Tests. [online]. [Accessed 19 May 2016]. Available from: <http://www.extremeprogramming.org/rules/unittests.html>
29. Pressman, Roger S. *Ingeniería de Software Un enfoque práctico*. [trad.] 601. 5. La Habana : Félix Varela, 2005. Vol. 2.
30. —. *Ingeniería de Software Un enfoque práctico*. 5. La Habana : Félix Varela, 2005. pág. 601. Vol. 1.
31. Sommerville, Ian. *Ingeniería de Software*. 7. Madrid : s.n., 2005. pág. 712.

32. Sampier, Roberto Hernández. Metodología de la investigación. La Habana : Félix Varela, 2008. pág. 475. Vol. 1.

33. —. Metodología de la investigación. La Habana : Félix Varela, 2008. pág. 475. Vol. 2.

ANEXO 1

Tabla 21: HU # 5 Listar tipo de notificación.

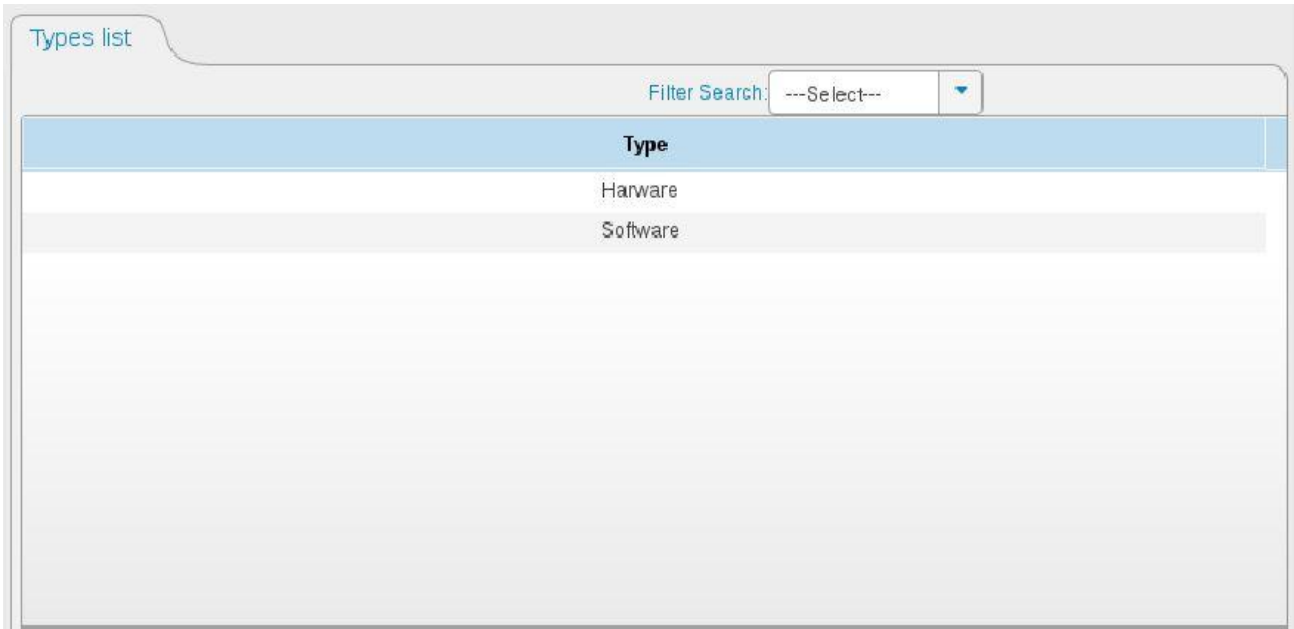
Historia de Usuario	
Número: 5	Usuario: Usuario
Nombre de historia: Listar tipo de notificación.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1/5	Iteración asignada: 2
Programadores responsables: Elaine de la Caridad Díaz Rodríguez, Víctor Raúl Ramos Linares	
Descripción: El usuario da clic en la opción "Types" y el sistema muestra una lista de todos los tipos de notificación creados por el usuario.	
Observaciones:	
Prototipo de Interfaz:	
 <p>The screenshot shows a web interface titled "Types list". At the top right, there is a "Filter Search:" label followed by a dropdown menu currently set to "---Select--". Below this is a table with a single column header "Type". The table contains two rows: "Harware" and "Software".</p>	

Tabla 22: HU # 6 Adicionar tipo de notificación.

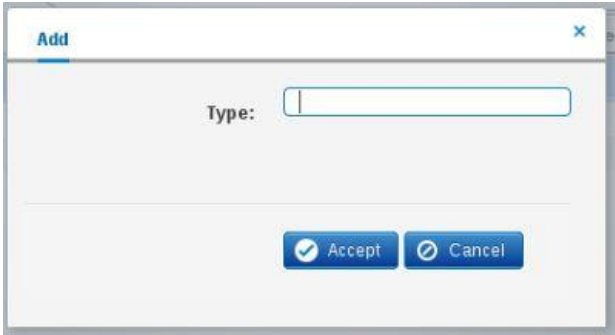
Historia de Usuario	
Número: 6	Usuario: Usuario
Nombre de historia: Adicionar tipo de notificación	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1/5	Iteración asignada: 1
Programadores responsables: Elaine de la Caridad Díaz Rodríguez, Víctor Raúl Ramos Linares	
<p>Descripción: El usuario da clic en la opción “add”, del menú “Types” y el sistema muestra una ventana emergente con el campo “Types”. Luego de introducido el nuevo tipo de notificación el usuario da clic en la opción “Accept” y el nuevo tipo es creado.</p>	
<p>Observaciones: El campo tipo de notificación es obligatorio.</p>	
<p>Prototipo de Interfaz:</p> <div style="text-align: center;">  </div>	

Tabla 23: HU # 7 Editar tipo de notificación.

Historia de Usuario	
Número: 7	Usuario: Usuario
Nombre de historia: Editar tipo de notificación.	

Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1/5	Iteración asignada: 2
Programadores responsables: Elaine de la Caridad Díaz Rodríguez, Víctor Raúl Ramos Linares	
Descripción: El usuario da clic en la opción “edit” del menú “Types” y el sistema muestra una ventana emergente donde permite modificar el tipo de notificación.	
Observaciones:	
Prototipo de Interfaz:	
	

Tabla 24: HU # 8 Eliminar tipo de notificación.

Historia de Usuario	
Número: 8	Usuario: Usuario
Nombre de historia: Eliminar tipo de notificación.	
Prioridad en negocio: Baja	Riesgo en desarrollo: Baja
Puntos estimados: 1/5	Iteración asignada: 3
Programadores responsables: Elaine de la Caridad Díaz Rodríguez, Víctor Raúl Ramos Linares	

Descripción: El usuario selecciona un tipo de notificación de los que se encuentra en “Types List” da clic en la opción “Delete” del menú “Types” y el sistema muestra una ventana para confirmar si se desea eliminar el tipo seleccionado. El usuario ratifica la opción de eliminar el tipo de notificación dando clic en el botón “Accept” y el tipo de notificación seleccionada es eliminada.

Observaciones:

Prototipo de Interfaz:

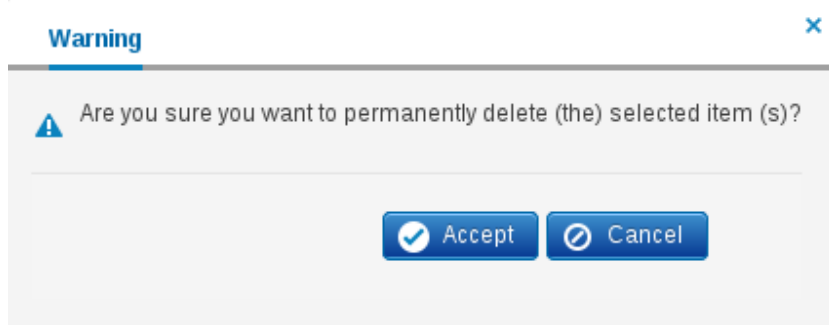


Tabla 25: HU # 9 Listar configuración de correo electrónico.

Historia de Usuario	
Número: 9	Usuario: Usuario
Nombre de historia: Listar configuración de correo electrónico	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 2/5	Iteración asignada: 2
Programadores responsables: Elaine de la Caridad Díaz Rodríguez, Víctor Raúl Ramos Linares	
Descripción: El usuario da clic en la opción “Mail” dentro del menú “Settings” y el sistema muestra una lista de todas las configuraciones de correo electrónico creadas por el usuario, con los campos: usuario, servidor SMTP, puerto, conexión segura (TTL) y el estado de dicha configuración (activa, o no activa).	
Observaciones:	

Prototipo de Interfaz:

User	Server SMTP	Port	TTL	Active
vrramos@estudiantes.uci	smtp.estudiantes.uci.cu	25	✓	✓

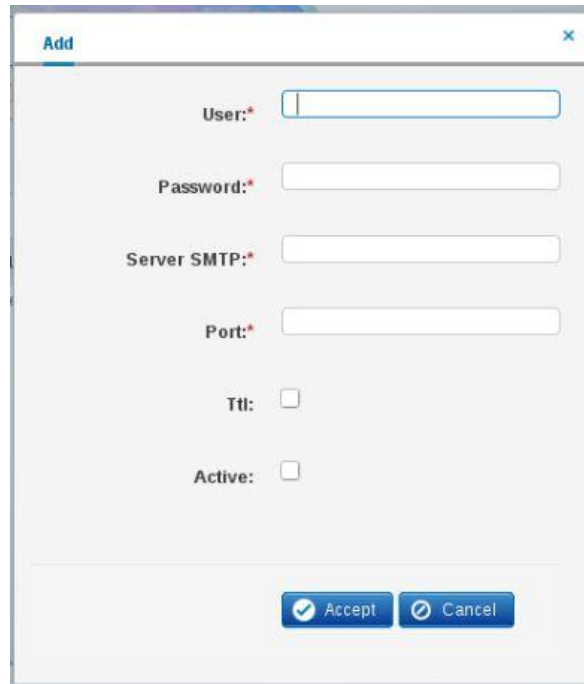
Tabla 26: HU # 10 Adicionar configuración de correo electrónico.

Historia de Usuario	
Número: 10	Usuario: Usuario
Nombre de historia: Adicionar configuración de correo electrónico	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 4/5	Iteración asignada: 1
Programadores responsables: Elaine de la Caridad Díaz Rodríguez, Víctor Raúl Ramos Linares	
Descripción: El usuario da clic en la opción “add” dentro del menú “Mail” y el sistema muestra una ventana emergente con los campos: usuario, contraseña, servidor SMTP y puerto; además permite seleccionar conexión segura (TTL) y el estado de la configuración (activa, o no activa). Luego de introducidos los datos de la nueva configuración de correo electrónico el usuario da clic en la opción “Accept” y la nueva configuración es creada. Esto ocurre siempre y cuando se valide que los datos	

introducidos por el usuario al crear la configuración de correo electrónico, sean los correctos y cumplan con los parámetros establecidos por el programador, ejemplo de esto es que no existan campos obligatorios en blanco.

Observaciones: Los campos usuario, contraseña, servidor SMTP y puerto son de carácter obligatorio.

Prototipo de Interfaz:



The image shows a software dialog box titled "Add" with a close button (X) in the top right corner. The dialog contains the following fields and controls:

- User:*** A text input field with a red asterisk indicating it is required.
- Password:*** A text input field with a red asterisk indicating it is required.
- Server SMTP:*** A text input field with a red asterisk indicating it is required.
- Port:*** A text input field with a red asterisk indicating it is required.
- Ttl:** A checkbox that is currently unchecked.
- Active:** A checkbox that is currently unchecked.

At the bottom of the dialog, there are two buttons: "Accept" (with a checkmark icon) and "Cancel" (with a close icon).

ANEXO 2

Subsistema: Gestión de notificaciones.

Tabla 27 CRC. Clase: Jabber.

Clase: Jabber	
Responsabilidad	Colaborador
Esta clase es la encargada de definir el modelo correspondiente a la configuración de la mensajería instantánea en la base de datos.	JabberSerializer

Tabla 28 CRC. Clase: JabberSerializer.

Clase: JabberSerializer	
Responsabilidad	Colaborador
Esta clase es la encargada de generar una representación apta para almacenamiento de los datos de la configuración de la mensajería instantánea, además de ser la encargada de obtener los datos desde la vista.	Jabber

Tabla 29 CRC. Clase: JabberMainMenu.

Clase: JabberMainMenu	
Responsabilidad	Colaborador
Esta clase es la encargada de definir la raíz visual de menú mensajería instantánea en la aplicación web.	ConfiguracionMainMenu JabberMenu

Tabla 30 CRC. Clase: JabberMenu.

Clase: JabberMenu	
Responsabilidad	Colaborador

Esta clase es la encargada de gestionar todas las funcionalidades correspondientes a la gestión de las notificaciones vía mensajería instantánea, mostrando la interfaz correspondientes a estas funcionalidades.	JabberMainMenu
---	----------------

Tabla 31 CRC. Clase: SMS.

Clase: SMS	
Responsabilidad	Colaborador
Esta clase es la encargada de definir el modelo correspondiente a la configuración del SMS en la base de datos.	SMSSerializer

Tabla 32 CRC. Clase: SMSSerializer.

Clase: SMSSerializer	
Responsabilidad	Colaborador
Esta clase es la encargada de generar una representación apta para almacenamiento de los datos de la configuración del SMS, además de ser la encargada de obtener los datos desde la vista.	SMS

Tabla 33 CRC. Clase: SMSMainMenu.

Clase: SMSMainMenu	
Responsabilidad	Colaborador
Esta clase es la encargada de definir la raíz visual de menú SMS en la aplicación web.	SMSMenu ConfiguracionMainMenu

ANEXO 3

Tabla 34: Tarea de Ingeniería # 5 Detalles de la notificación.

Tarea de Ingeniería	
Número tarea: 5	Número historia de usuario: 4
Nombre tarea: Detalles de la notificación.	
Tipo de tarea: Desarrollo	Puntos estimados: 1/5
Fecha inicio: 27/4/2016	Fecha fin: 27/4/2016
Programador Responsable: Elaine de la Caridad Díaz Rodríguez, Víctor Raúl Ramos Linares	
<p>Descripción: El usuario da clic en la opción detalles y la aplicación cliente consulta a la base de datos para obtener la información asociada a dicha notificación. Automáticamente el sistema muestra una ventana emergente con los datos referentes a la notificación seleccionada, los datos son: tipo de notificación, nombre(s) de lo(s) contacto(s) seleccionados, mensaje y muestra las vías de comunicación correo electrónico, mensajería instantánea, SMS y VoIP, resaltando por cual(es) se envió la notificación y el estado asociado a la misma.</p>	

Tabla 35: Tarea de Ingeniería # 6 Listar tipo de notificación.

Tarea de Ingeniería	
Número tarea: 6	Número historia de usuario: 5
Nombre tarea: Listar tipo de notificación.	
Tipo de tarea: Desarrollo	Puntos estimados: 1/5
Fecha inicio: 23/3/2016	Fecha fin: 23/3/2016
Programador Responsable: Elaine de la Caridad Díaz Rodríguez, Víctor Raúl Ramos Linares	
<p>Descripción: El usuario da clic en la opción “Types” y el sistema consulta a la base de datos para obtener todas las configuraciones de correo electrónico creadas por el usuario. Automáticamente muestra una lista de todos los tipos de notificación creados por el usuario.</p>	

Tabla 36: Tarea de Ingeniería # 7 Adicionar tipo de notificación.

Tarea de Ingeniería	
Número tarea: 7	Número historia de usuario: 6
Nombre tarea: Listar tipo de notificación.	
Tipo de tarea: Desarrollo	Puntos estimados: 1/5
Fecha inicio: 19/2/2016	Fecha fin: 19/2/2016
Programador Responsable: Elaine de la Caridad Díaz Rodríguez, Víctor Raúl Ramos Linares.	
<p>Descripción: El usuario da clic en la opción “add”, del menú “Notification List” y el sistema muestra una ventana emergente con el campo “Types”. Luego de introducido el nuevo tipo de notificación el usuario da clic en la opción “Accept”. Automáticamente el sistema se comunica con la base de datos para enviarle el nuevo tipo de notificación y esta se almacena.</p>	

Tabla 37: Tarea de Ingeniería # 8 Editar tipo de notificación.

Tarea de Ingeniería	
Número tarea: 8	Número historia de usuario: 7
Nombre tarea: Editar tipo de notificación.	
Tipo de tarea: Desarrollo	Puntos estimados: 2/5
Fecha inicio: 24/3/2016	Fecha fin: 25/3/2016
Programador Responsable: Elaine de la Caridad Díaz Rodríguez, Víctor Raúl Ramos Linares	
<p>Descripción: El usuario da clic en la opción “add”, del menú “Types” y el sistema se conecta con la base de datos para obtener la información referente al tipo de notificación seleccionado. Luego se muestra una ventana emergente con el campo “Types” listo para ser modificado. Después de introducido el nuevo tipo de notificación el usuario da clic en la opción “Accept” y automáticamente este nuevo tipo es creado y almacenado en la base de datos.</p>	

Tabla 38: Tarea de Ingeniería # 9 Eliminar tipo de notificación.

Tarea de Ingeniería

Número tarea: 9	Número historia de usuario: 8
Nombre tarea: Eliminar tipo de notificación.	
Tipo de tarea: Desarrollo	Puntos estimados: 1/5
Fecha inicio: 28/4/2016	Fecha fin: 28/4/2016
Programador Responsable: Elaine de la Caridad Díaz Rodríguez, Víctor Raúl Ramos Linares	
<p>Descripción: El usuario selecciona un tipo de notificación de los que se encuentra en “Types List” da clic en la opción “Delete” del menú “Types List” y el sistema muestra una ventana para confirmar si se desea eliminar el tipo de notificación seleccionado. El usuario ratifica la opción de eliminar el tipo de notificación dando clic en el botón “Accept” y el tipo de notificación seleccionada es eliminada automáticamente de la base de datos.</p>	

Tabla 39: Tarea de Ingeniería # 10 Listar configuración de correo electrónico.

Tarea de Ingeniería	
Número tarea: 10	Número historia de usuario: 9
Nombre tarea: Listar configuración de correo electrónico.	
Tipo de tarea: Desarrollo	Puntos estimados: 2/5
Fecha inicio: 28/3/2016	Fecha fin: 29/3/2016
Programador Responsable: Elaine de la Caridad Díaz Rodríguez, Víctor Raúl Ramos Linares	
<p>Descripción: El usuario da clic en la opción “Mail” dentro del menú “Settings” y el sistema consulta a la base de datos para obtener todas las configuraciones de correo electrónico creadas por un usuario. Luego muestra una lista de todas las configuraciones de correo electrónico existentes, con los campos: usuario, servidor SMTP, puerto, conexión segura (TTL) y el estado de dicha configuración (activa, o no activa).</p>	

Tabla 40: Tarea de Ingeniería # 11 Adicionar configuración de correo electrónico.

Tarea de Ingeniería

Número tarea: 11	Número historia de usuario: 10
Nombre tarea: Adicionar configuración de correo electrónico.	
Tipo de tarea: Desarrollo	Puntos estimados: 3/5
Fecha inicio: 22/2/2016	Fecha fin: 24/2/2016
Programador Responsable: Elaine de la Caridad Díaz Rodríguez, Víctor Raúl Ramos Linares	
<p>Descripción: El usuario da clic en la opción “add” dentro del menú “Mail” y el sistema muestra una ventana emergente con los campos: usuario, contraseña, servidor SMTP y puerto; además permite seleccionar conexión segura (TTL) y el estado de la configuración (activa, o no activa). Luego de introducido la nueva configuración de correo electrónico el usuario da clic en la opción “Accept” y automáticamente la nueva configuración es almacenada en la base de datos.</p>	

Tabla 41: Tarea de Ingeniería # 12 Validar configuración de correo electrónico.

Tarea de Ingeniería	
Número tarea: 12	Número historia de usuario: 10
Nombre tarea: Validar configuración de correo electrónico.	
Tipo de tarea: Desarrollo	Puntos estimados: 1/5
Fecha inicio: 25/2/2016	Fecha fin: 25/2/2016
Programador Responsable: Elaine de la Caridad Díaz Rodríguez, Víctor Raúl Ramos Linares	
<p>Descripción: El sistema se comunica con el servidor del servicio para verificar que los datos enviados por el cliente en la notificación sean válidos, de no ser así el sistema le muestra al usuario un mensaje de alerta mostrándole que datos no son válidos.</p>	